ON DETECTION AND USE OF REFLECTIONAL SYMMETRY IN COMPUTER VISION

Dipti Prasad Mukherjee
Indian Statistical Institute
203 B T Road
Calcutta 700035
India

Thesis submitted to Indian Statistical Institute
for partial fulfilment of the requirements for
the degree of Doctor of Philosophy

July 1994

To my grandparents and my mother

Abstract

The problems of detection and use of reflectional symmetry in the images of planar shape contours are studied. Symmetry, in general, provides important shape representation cues, some of which we utilise here, to acquire viewpoint information and, towards model based shape matching. We concentrate on local reflectional symmetries of smoothly curved planar objects, though the methods are equally applicable to polygonal objects; even this could be extended to certain three-dimensional shapes and for other object relations such as rotational symmetry.

Under the affine or perspective approximation to image projection, properties of geometric invariance are used to find (reflectional) symmetric contour pairs. Local symmetry line is detected from the transformation matrix (responsible for inter-symmetric-curve reflection) of symmetric contour pair. The key result we establish here, is that due to bilateral symmetry constraint, the image transformation between the symmetric contour pair is a subset of affine or projective transformation, as the case of image projection may be. The goal of the analysis, at this stage, is to remove the imaged distortion (perspective "fanning" and affine "skewing") due to the affine or perspective approximation to image projection. This is to recover the shape of the object up to similarity transform (rotation, translation and isotropic scaling) ambiguity. Symmetry constraint is utilised to find the "back-projection" matrix responsible for image distortion. This image-to-object transformation (up to similarity ambiguity) is used to determine the object plane orientations relative to the camera and, to test for non-co-planarity amongst a collection of objects.

As we establish algorithms for obtaining back-projected images from the "skewed" scene, we concentrate on the wave diffusion algorithm which gives *almost* all the *perceptually relevant* symmetry set for planar shapes up to similarity transform. We study this symmetry generation process and present the parallel implementation of this methodology on a transputer network. This is to overcome the basic drawback of wave diffusion process, *viz.* the high processing time. An attractive alternative, almost instantaneous, for symmetry set generation, we call *normal transform*, similar to the wave propagation mechanism is introduced.

Symmetry set of a planar shape, generated using wave diffusion process, is used for model based symmetry set matching. Symmetry sets of the image and the models are transformed to a *representative frame* to take care of images of objects up to similarity transform. The structural and relational matching constraints between the symmetry set of an image and the symmetry sets of models, combined together, are incorporated in an energy function. *Simulated annealing* technique is used to minimize this energy function to find the best match.

The thesis is concluded by a summary indicating the future scope of research.

Acknowledgement

I am grateful to my supervisor Prof. D. Dutta Majumder for having introduced me to the exciting field of Computer Vision. I am fortunate to have had the opportunity to share some of his great knowledge in the field, as well as his never-ending enthusiasm for it. Needless to say, the continuous encouragement and competent advice he has provided is invaluable.

I thank all of the members of the KBCS group, who are my colleagues but above all my friends, for making our group such an enjoyable and inspiring environment to work in. Many have helped improve the manuscript in one way or the other, most notably Bhabatosh Chanda, S E Sarma, Amita Pal, Dilip Banerjee, Amarnath Gupta, Suash Deb and B Uma Shankar. I thank Jayanta Basak and Nikhil Pal for some of their thoughtful suggestions particularly for chapter six. Special thanks are due to Profs. J. Das and S. K. Pal for their encouragements. I thank Tirtha Prasad Mukherjee for helping me improving the English of the manuscript.

Most of the ideas described in this thesis were cultivated and given the present form during my visit to the Robotics Research Group in University of Oxford, UK as an UNDP fellow. I thank UNDP to make my visit possible. Prof. J M Brady had supervised my work at Oxford. I am grateful to him for his technical guidance, valuable suggestions and critical review of parts of this thesis. There are hardly any words to express my gratitude to Andrew Zisserman who has come up with most of the stimulating ideas particularly for chapters three and four. He has patiently helped me in teaching the basics of geometric invariance enduring my endless emails. I thank you Andrew for all your help. I thank Andrew Blake for many a helpful suggestions. I must mention the names of some of my colleagues at Oxford who had helped some way or the other. I thank Paul Beardsley, Philip McLaughlin, Simon Turner, Charlie Rothwell, Mike Taylor, Fang Le Du, Roger Fawcett for all their helps.

During my association with the KBCS group at the Indian Statistical Institute, Calcutta, I had the opportunity to discuss some of my sketchy ideas with the authorities in computer vision. I particularly thank Profs. Olivier Faugeras and Thomas Binford who had helped towards improving some parts of the manuscript.

Finally, I would like to express my gratitude for the support and encouragement I have received from my family, my parents, my brother and my wife Santana, particularly for her patience and tolerance for the last three years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The representation and recognition of shapes by computer has numerous applications: as a step towards automating processes such as inspection; acquisition of objects from a conveyer belt or container; reconnaissance; and, navigation by an autonomous system. Also, such work can potentially contribute to understanding one of the most exquisite and effortless of human competences: the rapid recognition of familiar shapes even when they are partially occluded by others, when their surface colour or texture is unfamiliar, and when they are viewed from a wide range of vantage points. From a very early age, we can learn new classes of shapes, learn to discriminate subclasses, and then mobilise those new representations to effect recognition.

"... we arrived at the idea of a sequence of representations, starting with descriptions that could be obtained straight from an image but that are carefully designed to facilitate the subsequent recovery of gradually more objective, physical properties ..." [43]. Understanding how to represent and recognise shapes has, however, proved to be a remarkably difficult task, both for computer vision and for perceptual psychology. So much so, that the current state of the art is that only limited classes of shapes can be recognised reliably from a limited range of poses. To date, perceptual psychology has been of limited usefulness, for though theories of shape abound [6,18] they are too vaguely formulated to be implemented in a computer recognition system. On the other hand, most computer vision approaches to shape [3,44] have either emphasised gross shape characteristics (for example, low-

order moments or the first few coefficients of the Fourier or other transform of the contour function of the shape), or have relied on highly localised features, such as an estimate of points of high curvature along the bounding contour. Gross shape representations [29] have insufficient discriminatory power and are sensitive to occlusion, while very local representations are subject to measurement noise that is unavoidable in practice.

The most advanced representation and recognition techniques developed to date in computer vision have explored representations intermediate between these extremes, and have exploited one or more of: relational constraints between parts of a shape, prior models, symmetry properties of the shape, or affine/projective invariance. The relational constraints based approach relies upon precisely known algebraic relationships between different parts of a shape and has only been explored for the case of polyhedra, albeit classes of polyhedra defined parametrically [34,58]. Model-based recognition, even for non-polyhedral shapes, has enjoyed some success [34,42,58]. Over the past thirty years, researchers in computer vision have explored a number of aspects of symmetry to generate symbolic representations of shape, culminating in the systems reported by Blum and Nagel [10], Connell and Brady [17] and, more recently by Rom and Medioni [59]. The importance of invariance has been recognized since the origin of the field in the 1960s. The fundamental difficulty in recognizing objects from images is that the appearance of a shape depends on viewpoint (position and orientation of the camera). Geometric invariance generates important shape "signature" and recognition cue which remain unchanged under an appropriate class of image transformation. A useful collection is provided in [53]. The present work is a contribution combining these approaches primarily focusing on symmetry properties of a shape.

1.1 Shape from symmetry

Many important classes of shapes, from faces and leaves through to manufactured items such as many stamped metal parts and profiles of aeroplanes and buildings exhibit one or more symmetries. We, in fact, remember and enjoy symmetrical objects better than asymmetrical ones. There is a wealth of literature, spanning more than hundred years, discussing the mechanisms by which humans detect symmetry (see [22]). Within the plane, symmetry comprises reflections, rotations, and

translations, in which latter case symmetry corresponds to pattern repetition. The essential idea of a symmetry is a motion [26]: "suppose you have an object and pick it up, move it around and set it down. If it is impossible to distinguish between the object in its original and final positions, we say that it has a symmetry." The line of thinking encapsulated by the quotation leads inexorably to modeling symmetry using the operations of group theory [77], a point to which we return when discussing affine/projective invariant representations of shape. However, requiring the transformed object to be "impossible to distinguish" from the original is far too restrictive both for computer vision and for human perception. Real objects such as faces, pears, wrenches, and the outlines of fish [72] are only *approximately* symmetric, and, more significantly, the shape only exhibits symmetries locally between segments of a shape or pattern.

1.1.1 Detecting symmetries

Distinctive from the earlier psychological studies, the first exploration of local support for symmetry, using computational methods, was by Blum [9] in his study of representations of biological shapes to effect recognition, determine abnormalities, and monitor growth. Blum proposed an elegant "grassfire" method to recover *perceptually relevant* symmetry axes of a shape, an issue on which we concentrate when discussing the parallel implementation of the algorithm in chapter 5. Defining symmetry is not quite as simple as it seems. The symmetries we consider "correct" or *perceptually significant* for a square depend on its orientation: compare the squares and symmetries of figures 1.1 (a) and (b). The full symmetry for a square must therefore be as shown in figure 1.1 (c).

The idea of local symmetry was put on a more solid mathematical footing by Giblin and Brassett [32], who defined the *symmetry set* (SS) of a shape as the locus of the centres of all circles bi-tangent to a shape's bounding contour. A number of algorithms have been developed for computing the loci of local symmetries of shapes, particularly reflectional [11,65] and rotational symmetries [27]. They have been demonstrated to work reliably on a range of shapes, generating representations useful for recognition. Rosenfeld [60] provides a lucid account of the differences between Blum's [9], Brook's [14] and Brady's [11] definitions for symmetry generation. A more recent paper by Ponce [56] gives further comparisons. Eades [21] surveys several symmetry finding algorithms for pattern classes commonly studied in com-

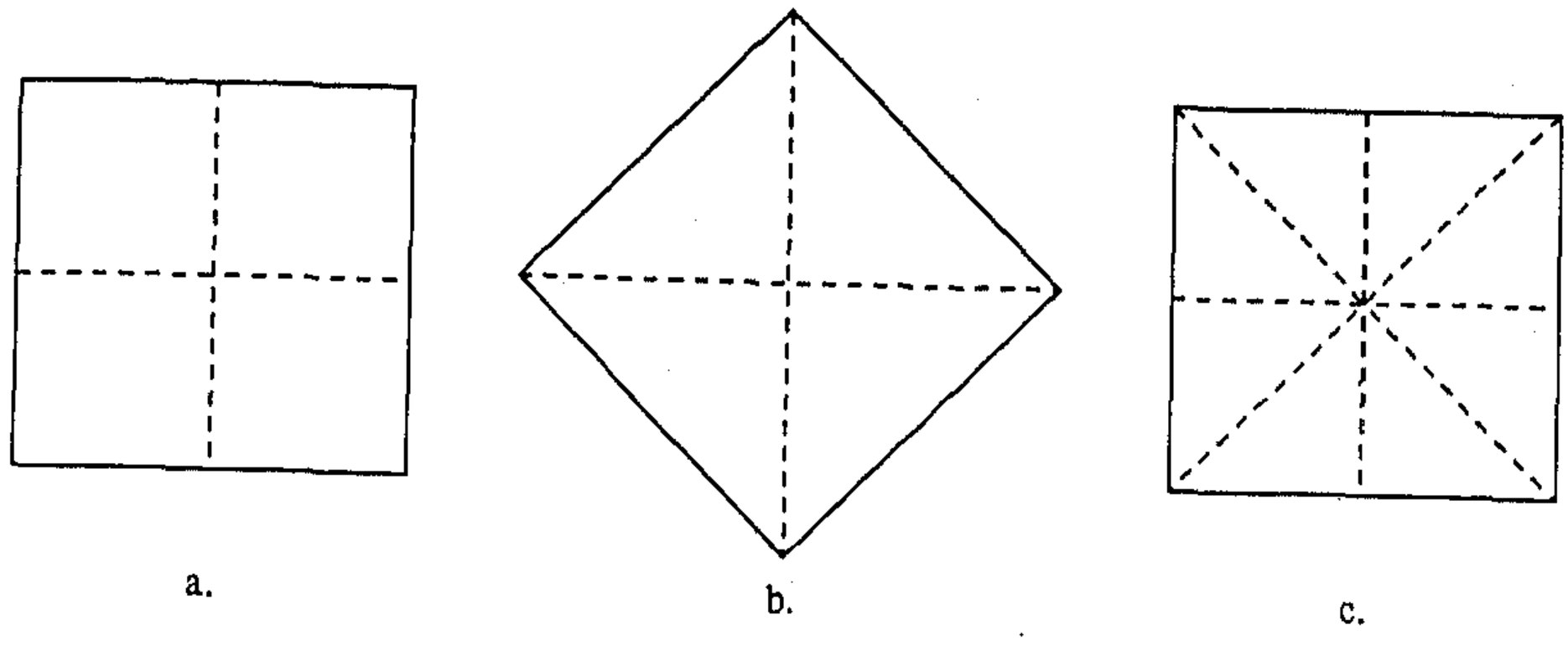

Figure 1.1: The "obvious" symmetries (dotted lines) for a square (a) and a diamond (b). In fact the two shapes are identical; the correct symmetries of a square are shown in (c).

putational geometry, graphics and pattern recognition: for example, finite set of points, polyhedra, finite sets of line segments, and planar subdivisions.

However, all such algorithms and representations share a severe limitation: symmetry is not preserved under skew, corresponding to the shape being viewed other than in a fronto- parallel plane. Simply stated, symmetry axes computed in an image of a shape taken from a non-fronto-parallel vantage point are not in general the transformed fronto-parallel symmetry axes. Despite this mathematical inconvenience, "skewed symmetries" (that is, reflectional symmetries viewed from a non-fronto-parallel vantage point) such as those shown in figure 1.2 strongly suggest actual symmetries and constrain the plane in which they are perceived to lie. The mathematical fact that the skew symmetry may be an accident of projection is evidently discounted. Indeed, Wagemans [75] has recently provided evidence that skewed symmetry is a *nonaccidental*[1] [78,42] property of a shape that the human visual system exploits. Kanade [37] was the first to analyse mathematically symmetries skewed by image projection, and proposed heuristics to interpret a skew symmetry as a real symmetry viewed from some (unknown) direction, which he represented using gradient space. A moment-based approach [30] was proposed by Friedberg. Kelvin Yuen developed algorithms [68] to detect skew symmetry and *projected rotational symmetry*. Van Gool *et al.* used arc length space (ALS) [33] to extract affine

[1]Critical information is unlikely to be a consequence of an accident of viewpoint.

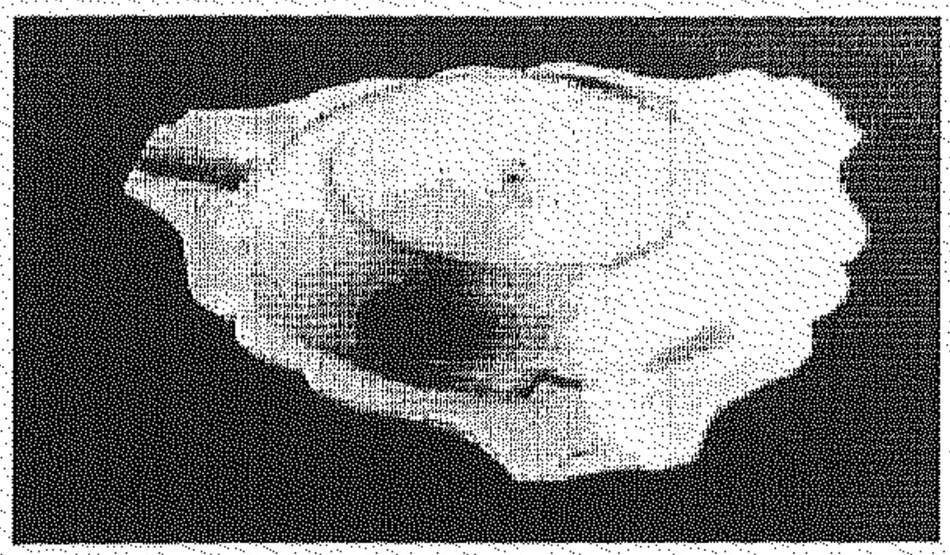

Figure 1.2: Example of skewed symmetry. The object is an inverted ash tray and has three bilateral symmetries, but the symmetry axes are not orthogonal. In each case the symmetry is global, and *on the object* lines between points related by the reflectional symmetry are orthogonal to the axis of reflection. However, in the image these lines and the axis are not orthogonal, in general, but skewed.

symmetry information using semi-differential invariants.

Notwithstanding the cogency of skewed symmetric shapes, and the information they seem to present about the pose (rotation and translation relative to the camera) of the shape, the fact remains that symmetry is not invariant under projection. This has recently persuaded a number of authors to abandon temporarily such representations to explore projective and affine invariants of two-dimensional shapes [53,62,76,2]. Like symmetry, invariance is also normally formalised using the mathematics of group theory, particularly Lie group theory [53]. Systems have been developed exploiting a variety of invariants to recognise overlapped shapes viewed from a variety of poses. This substantial progress has been bought at the cost of regressing shape representations, now in a "canonical frame", to descriptions invariant to pose.

Although not strictly part of this study of geometrical symmetry, the concept of gray level symmetry has attracted some interest. Crowley and Parker [19] have proposed algorithms for computing gray level symmetries. Thornham *et al.* [73] have used "centre of gravity" (COG) filter of gray levels to extract geometrical symmetry information.

1.1.2 Using symmetries

Symmetry axes have various applications in computer vision and related fields. This section will outline some of them, but by no means is the list exhaustive.

Shape descriptions

- Boundary reconstruction
  Blum [9] observed that his set of "sym-points" (Symmetry Axis Transform (SAT) points) with their associated "sym-distances" (radii of appropriate circles) formed a complete description of shape; it is possible to completely reconstruct a shape's boundary using the envelope of "sym-distances". He did not, however, propose a way of using this in a practical system.

- Symmetric axis features
  Blum [9] used, what he called "A-morphology" (axis morphology[2]) to describe a shape in terms of the behaviour of its SAT axes.

  Blum and Nagel [10] extended the idea of shape primitives into a description which segmented the shape into simplified segments based on axis properties, segment end and join, then formed a directed graph using weighting measures for each segment.

- Symbolic description
  The system in [17] builds semantic network descriptions of shapes based on "smoothed local symmetry" (SLS) [11] for *learning* planar shapes. Shape segmentation used various shape parameters such as length, aspect ratio, area and orientation.

- *Hierarchical descriptions*
  Pizer, Oliver and Bloomberg [55] used the SAT to generate a hierarchical shape descriptions at different scales of resolution, by tracking the disappearance of SAT branches as the resolution is reduced.

  Rom and Medioni [59] use both region and contour based information for hierarchical decomposition and axial representation of planar shape. They address the issue of local versus global information and the intuitive notion of part.

[2] As opposed to "B-morphology", boundary morphology

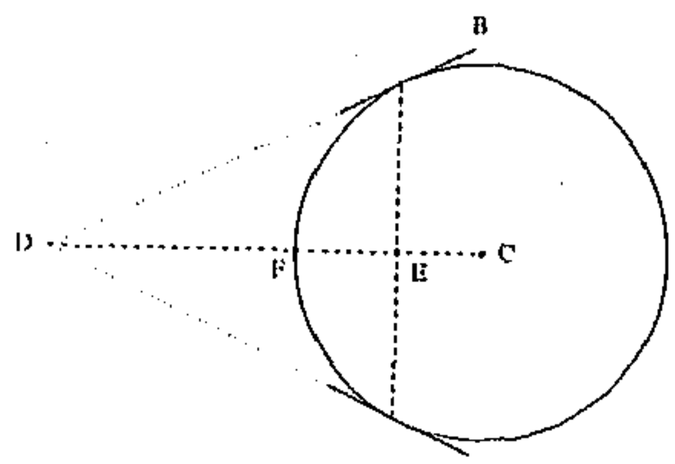

Figure 1.3: For a pair of edgels A and B, the point C represents both Symmetry Set [32] and SAT [9] points whereas E is the SLS point [11]. Point F represents PISA [40] point on the circumference of the circle.

Morphogenesis

Leyton [40] uses symmetry information to attempt to infer processes that could have acted on a basic shape to achieve its present form. This analysis is appropriately called *Process-Inferring Symmetry Analysis* (PISA). This appears to have two major uses - shape description and inference of process history.

For a pair of edgels A and B, the different symmetric points due to SAT, SLS, PISA and SS are illustrated in figure 1.3.

Recovering 3D shapes

For a variety of 3D objects, Ulupinar and Nevatia [74] have shown three types of symmetry: skewed, parallel and line-convergent symmetries give significant information about the surface. The recovery of shape of the surface is, however, limited to objects consisting of zero-Gaussian curvature surfaces. In [45], co-curvilinearity and symmetry are used as perceptual grouping processes to generate the collated features including curves and ribbons. These collated features including Transformationally Invariant Symmetry Analysis (TISA) segment scenes and describe the 2D shapes of surfaces.

Shape back-projection

Recently attempts [52,51] have been made to extract symmetry information of affine and projective images of planar shapes using geometric invariance properties of affine and projective transformations. Back-projection is defined as the transformation from image to object plane after removing the perspective and affine distortion of the imaged object due to non-fronto-parallel viewing. The symmetry constraint is utilised to back-project the image modulo similarity transform (i.e. up to rotation, translation and isotropic scaling ambiguity) and to calibrate the camera. These would be a major part of deliberations of this thesis. A similar approach has been explored by Fawcett *et al.* [25] to recover structure modulo similarity from an affine view of a 3D point set with one or two bilateral symmetries.

Other uses

- Shape recognition
  Mukherjee and Dutta Majumder [50] have proposed a shape symmetry matching technique given a model base of symmetry sets of shapes described in the *representative frame*. The key idea is to formulate the matching process as minimization of an energy function.

- Skeletonisation
  Symmetry axes are obviously good at skeletonising [9] certain shapes. A number of symmetry detection procedure, described above, are successfully utilized to skeletonise a shape reducing the amount of information needed to store the shape, without reducing the ability to recognise it.

- Path planning
  Canny and Donald [16] have used the Voronoi diagram in robot path planning, selecting the natural path which, in simple terms, keep the robot as far away from all obstacles as possible.

1.2 The present work

After an introduction of the basic geometric framework in chapter 2, the present thesis is broadly divided into three parts. In the first part, spanning chapters 3 and 4, we are interested to recover shapes modulo similarity from the images under affine and projective transformations. The contribution is a step forward reconciling symmetry and invariance. In the second part, in chapter 5, we concentrate on the efficient detection of symmetry set for shapes under similarity transform. The contribution is the parallel implementation of a *high-level* vision algorithm. Lastly, we show a novel application of *perceptually relevant* symmetry set in chapter 6 for shape matching, the contribution being the use of a smart minimization technique towards model based object recognition.

Henceforth, by symmetry, we mean local reflectional symmetry, unless mentioned otherwise. We concentrate on symmetries of smoothly curved planar objects, though the methods are equally applicable to polygonal objects, and even could be extended to certain three-dimensional shapes and for other object relations such as rotational symmetry.

As stated earlier, chapter 2 briefly provides a consistent framework of the image transformation and parameters for image projection which we utilize in subsequent chapters.

Suppose then that a planar object has a bilateral symmetry; how does this constrain its image projection? If the two "sides" of the contour have a mirror symmetry, then one can be transformed onto the other by a reflection. A reflection is a particular type of projective transformation - the most general planar object to image transformation. However, the key result we establish in the subsequent chapters is that because of bilateral symmetry constraints, this image transformation is a subset of planar projective group. We assume that the projection between the object and image planes can also be approximated by a projective transformation. Our aim is to evaluate this transformation (perspective and affine distortion) for image to object back-projection ("unskewing") modulo a similarity transform. We will show that this information can usefully be used to determine the slant and tilt (orientation) of the object plane with respect to camera position and for a test for non-co-planarity amongst a collection of objects.

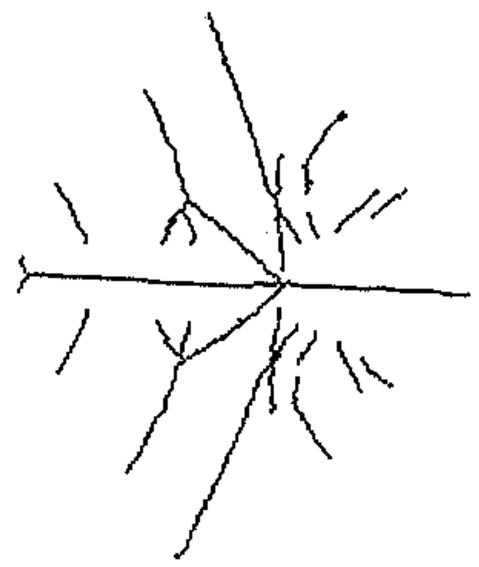

Figure 1.4: Symmetry set of the edge image of DC10 aircraft after 95 iterations of wave diffusion process followed by non-maximal suppression.

In chapter 3, we assume the image transformation to be a subgroup of planar projective group - planar affine group. We detect symmetry [52] in the affine images of smoothly curved planar objects using *affine invariants*. We utilize this affine symmetry [52] to back-project the image, calibrate the camera and test the planarity conditions of a collection of objects.

Chapter 4 is the direct extension [51] of chapter 3, the image transformation assumed being planar projective group.

As we establish algorithms for obtaining unskewed images from the "skewed" scene, we concentrate on a symmetry finding algorithm which gives *almost* all the *perceptually relevant* symmetry set for planar shapes up to similarity transform. The wave diffusion algorithm, originally thought by Blum [9] ("grassfire" technique), implemented by Scott *et al.* [65] provides such an alternative. An example is shown in figure 1.4 where symmetry set of the edge image of DC10 aircraft is obtained following [65].

However, a significant drawback of the symmetry set evaluation of a 2D shape using the wave diffusion process is its slow execution time caused, in large part, by the diffusion step. In chapter 5, we first recall the need for a diffusion step followed by a parallel implementation [49] of the wave diffusion algorithm on a transputer network. A faster alternative approach to detect the symmetry set, which is termed as *normal transform* and which is similar to the wave propagation mechanism is detailed [48].

Chapter 6 describes the matching process [50] of symmetry set of a planar contour, generated using wave diffusion algorithm, with candidate models of a model base, consisting of symmetry sets of models. The structural and relational matching constraints are incorporated in an energy function and *simulated annealing* technique is used to minimize this function to find the best match.

Chapter 7 concludes the thesis with a summary and a discussion on future direction.

Chapter 2

Geometric Framework

Before we proceed further, we detail the fundamentals of imaging and surface geometry. Naturally, the basic issues have been discussed many times before in the vision literature. As a consequence, most of the results to be presented in this chapter are well-known, although not necessarily in the form in which they will be given here. General references are too numerous to list; however, a thorough and lucid discussion in the context of shape from surface markings, which we have tried to follow, is given by Garding [31].

The analysis is based on the familiar pinhole model of image formation. The transformations [53,23] described here are from plane to plane *collineations*. Projection from planar surface model is considered because of its abundance in man-made environment. Also, many surfaces can be considered *locally* planar and their mathematical analyses are much simpler compared to any other class of surfaces.

This chapter provides a sufficient background for the rest of the thesis and chapters 3 and 4, in particular. The chapter is organised as follows: We discuss imaging geometry in section 2.1. The issues relating to image transformations are dealt in section 2.2 followed by discussions.

Notation Throughout the thesis, we adopt the notation that vectors are written in bold font (*e.g.* $\mathbf{x}$) and matrix are written in typewriter font (*e.g.* $\mathtt{X}$).

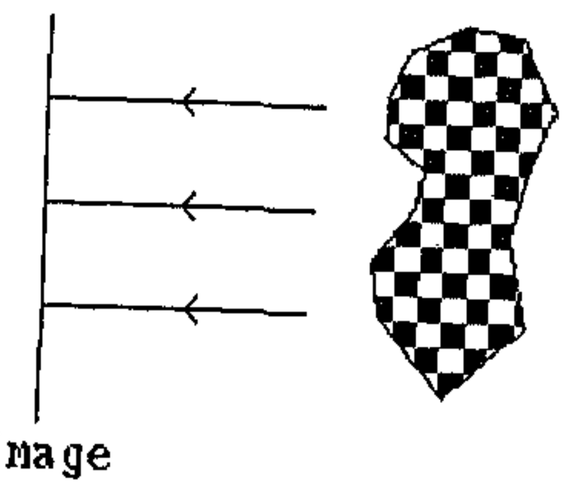

Figure 2.1: The *orthographic* projection model with visual rays all perpendicular to the image plane.

2.1 Imaging geometry

The two imaging models most commonly used are *orthographic* and *perspective* projection. Contrary to our visual experience, in *orthographic* projection, the size of the image of an object is independent of the distance from the image to the object. Figure 2.1 shows the *orthographic* projection model where the visual rays are all perpendicular to the image plane. For smaller viewing angle, this is a good approximation of *perspective* projection, where the visual rays converge onto a common point, the *focal* point. However, this approximation, unfortunately, is not strictly true for real scenes, inspite of the mathematical simplicity of *orthographic* projection.

On the other hand, a common viewing situation, which cannot be modeled by *orthographic* projection, is that the ground plane will occupy half of a *perspective* observer's field of view, given his central line of sight being parallel to the infinite ground plane. The image formed by *perspective* projection is shown in figure 2.2. Thus, *horizon* can never be obtained by *orthographic* projection which is purely a *perspective* phenomenon. *Orthographic* projection may be a reasonable approximation when we have a priori knowledge that the planar surface is approximately parallel to the image plane.

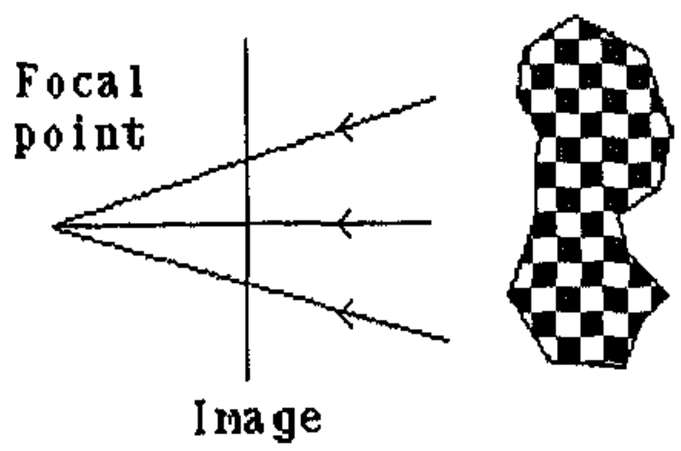

Figure 2.2: The *perspective* projection model where visual rays converge to the *focal* point.

2.1.1 Projection from a planar surface

Refer to figure 2.3. The projection of planar surface S onto the image plane is analysed [31] in the *pinhole camera* model. Orientation of S is given by *slant* angle σ and *tilt* angle τ. XYZ and xy are the camera and the image coordinate systems respectively. The local coordinate system of S is given by uvw where $w = 0$ as we are dealing only with 2D objects. The w axis is parallel to the surface normal $\mathbf{n}$ of S pointing away from the image. The slant σ ($\in [0, \pi]$) is the angle between $\mathbf{n}$ and the optic axis (Z) while tilt τ ($\in [0, 2\pi]$) is the angle between the parallel projection of $\mathbf{n}$ onto the image plane and the x axis of the image coordinate system. Surface normal at the origin O_{uvw} of S passes through O_{XYZ}. The distance between these origins is given by δ. Stevens [71] has compared the human perception of surface representation with the parametric surface representation using (σ, τ, δ).

To enumerate the transformation from the coordinate system uvw to XYZ:

1. Parallel translation along $\mathbf{n}$ to the coordinate system $u'v'w'$ whose origin coincides with O_{XYZ}:

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} u \\ v \\ w + \delta \end{pmatrix} \tag{2.1}$$

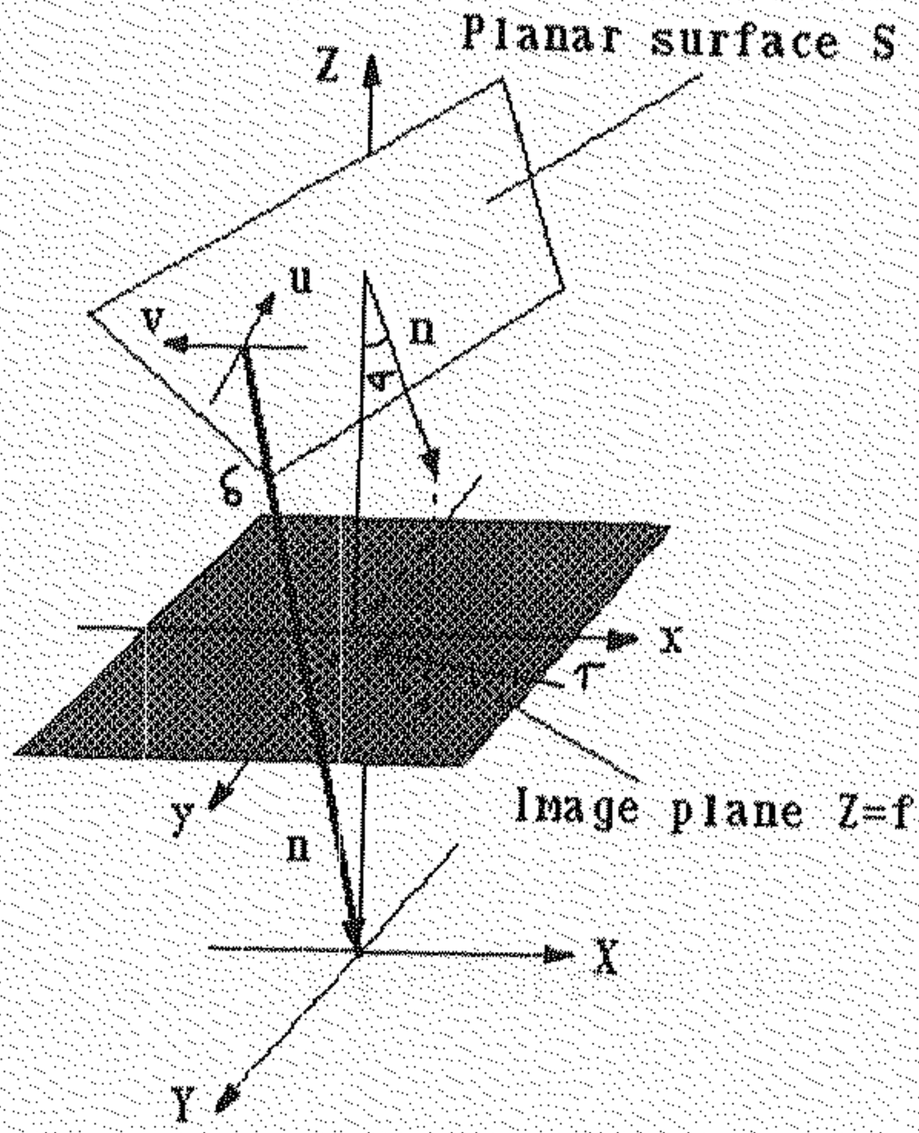

Figure 2.3: Projection of planar surface S onto the image plane xy. Surface orientations are given by *slant* angle σ and *tilt* angle τ.

2. Rotate σ around the v' axis to make the plane $w'' = 0$ coincide with $Z = 0$:

$$\begin{pmatrix} u'' \\ v'' \\ w'' \end{pmatrix} = \begin{pmatrix} \cos\sigma & 0 & -\sin\sigma \\ 0 & 1 & 0 \\ \sin\sigma & 0 & \cos\sigma \end{pmatrix} \begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} \quad (2.2)$$

3. Rotate τ around the w'' axis:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \cos\tau & -\sin\tau & 0 \\ \sin\tau & \cos\tau & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u'' \\ v'' \\ w'' \end{pmatrix} \quad (2.3)$$

The complete coordinate transformation is obtained by concatenation of the preceding three steps.

In case of *perspective* projection, the image plane xy is the plane at $Z = f$ where f is the focal length, x and y axes being parallel to X and Y axes respectively. For *orthographic* projection the image plane is at $Z = 0$ coinciding x and y with X and Y axes respectively. Since, translating the image plane along the optic axis (Z) does not change the image for *orthography*, we can conveniently choose $\delta = 0$ bringing O_{uvw} to O_{XYZ}.

2.2 Plane to plane transformations

The plane to plane transformations describe world model to image mappings and vice versa. We begin their descriptions starting from the plane projective group. These transformations can be represented by a *group* since they satisfy the group axioms of: *closure*, *identity*, *inverse* and *associativity*. While there exists more general transformations [53] than projectivities, the detail discussion of those is beyond the scope of this thesis.

The plane projective group A projective transformation or *projectivity*, from one projective plane Π, to another, π, is a non-singular 3×3 matrix acting on homogeneous coordinates.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \quad (2.4)$$

or

$$\mathbf{x} = \mathrm{T}\mathbf{X} \tag{2.5}$$

The transformation matrix T has eight degrees of freedom because only the ratio of homogeneous coordinates is significant and there are 8 ratios among 9 elements of T.

Properties like concurrency, collinearity, order of contact (intersection, tangency, inflections), tangent discontinuities and cusps and cross-ratio are preserved under projective transformation [53].

The plane affine group In case of affine transformation the matrix T, as in equation (2.5), takes the form:

$$\mathrm{T} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & t_{33} \end{pmatrix} \tag{2.6}$$

Affine transformation has six degrees of freedom and is equivalent to the combined effects of translation, rotation, isotropic scaling and shear (non-uniform scaling in some direction).

Properties like parallelism, ratio of lengths of collinear or parallel segments (e.g. mid-points), ratio of areas, linear combination of vectors are invariant under affine transformation [53].

The plane similarity group This is a specialization of the affine transformation without shear and is equivalent to a Euclidean transformation composed with an isotropic scaling. This has four degrees of freedom and occurs when the world plane is parallel to the image plane i.e. under *fronto-parallel* viewing. Ratio of lengths, angles are preserved under plane similarity transform [53].

The plane Euclidean group The Euclidean transformation matrix is shown in equation (2.7). Here the top 2×2 sub-matrix of T is a rotation matrix and $\mathbf{t} = (t_1, t_2)^t$ is a translation vector. It has three degrees of freedom.

$$\mathrm{T} = \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

Lengths, angles, areas are preserved under the Euclidean transform [53].

2.3 Discussions

The motivation for using a specific transformation geometry depends on the physical imaging process. While the plane projective group is a *perspective* phenomenon, the transformation due to affine group is the effect of a special type of *weak perspective* where the parallel lines in the object remain parallel in the image. The similarity or Euclidean transformation is due to orthographic projection for *fronto-parallel* viewing.

There is a strict hierarchy of plane to plane transformations beginning from the plane projective group. Each group inherits the invariances of the more general transformations of the preceding groups, but also have extra invariances. The hierarchy and the range of transformations are shown in figure 2.4.

Chapter 3 deals with the *affine* images where the invariances of affine group are used to detect the symmetry. It is easy to show, for example using a Taylor's series expansion of the projection equations of a pinhole camera, that affine group is a good approximation provided the field of view is small, and the range of depths encompassing the object is at least an order of magnitude less than the distance of the object from the camera.

Chapter 4 is the extension of the ideas of chapter 3 to a planar projective group. However, in both the cases, the images are back-projected modulo a similarity transform.

The symmetry detection methodologies presented in chapter 5 and its application in chapter 6 concentrate on planar shapes under similarity transform.

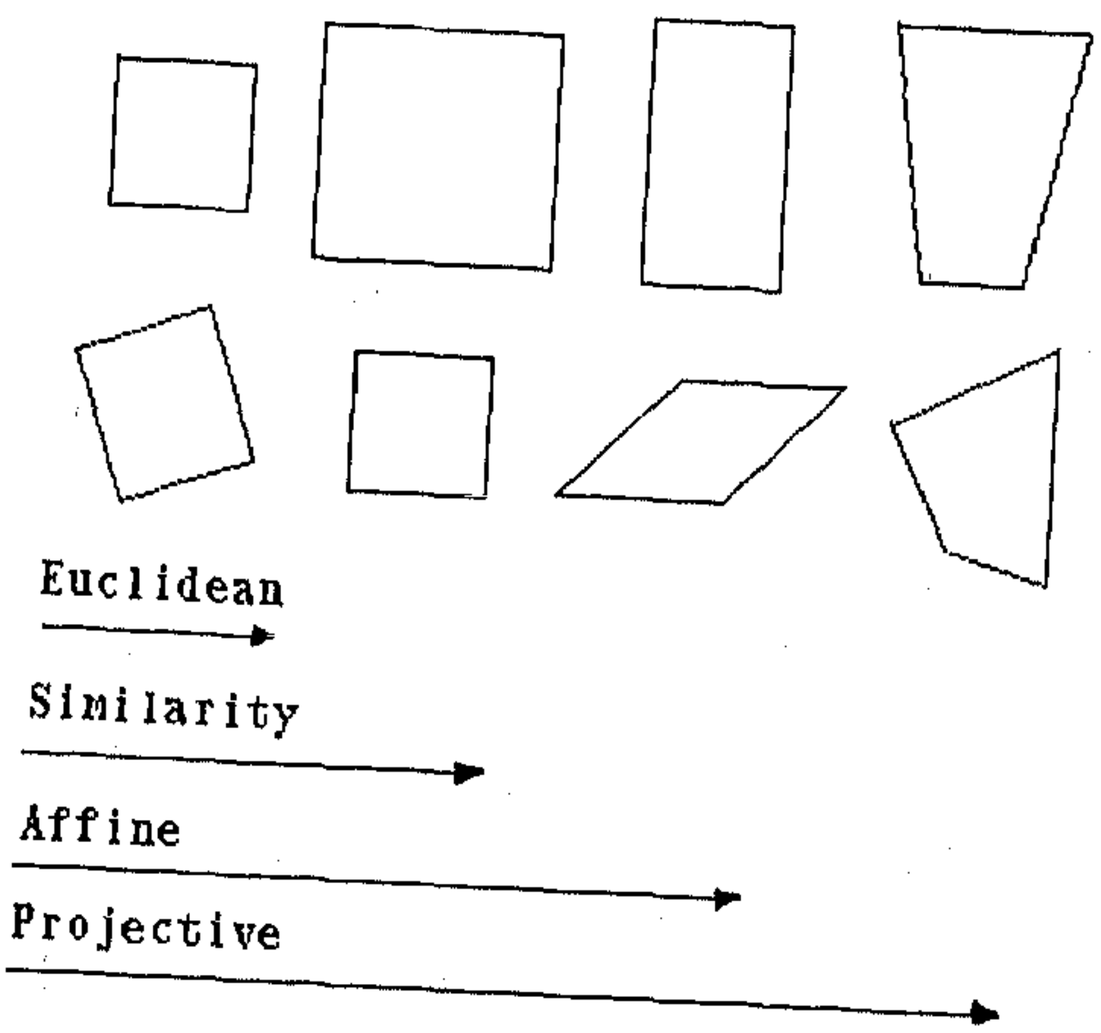

Figure 2.4: The hierarchy of plane to plane transformations ranging from Euclidean to plane projective group.

Chapter 3

Symmetry in Affine Images

3.1 Introduction

We investigate the constraints placed on the image projection of a planar object with local reflectional symmetry. Under the affine approximation to projection, we demonstrate an efficient algorithm for detecting and verifying symmetries despite the distorting effects of image skewing. The symmetries are utilised for three distinct tasks: Firstly, image to object back-projection modulo a similarity transformation; secondly, determining slant and tilt of the object plane; thirdly, as a test for non-coplanarity amongst a collection of objects. These results are illustrated throughout with examples from images of real scenes.

We assume that the projection between the object and the image plane is approximated by an affine transform. If the two "sides" of a contour have a mirror symmetry, then one can be superimposed onto the other by a reflection. A reflection is a particular type of affine transformation. Since affine transformations form a group, *the transformation between the two sides of the contour in the image* is also affine. This immediately provides an algorithm, albeit one that is computationally expensive, for detecting possible symmetries: if two image contours can be mapped onto each other by an affine transformation (six degrees of freedom), then the object could have had a reflectional symmetry. However, a key result of the present chapter is given in section 3.2, where it is shown that the image transformation is actually a subset of the affine transformations with only *three* degrees of freedom. This,

and the use of affine index functions (section 3.3), is used to develop an efficient algorithm for detecting and verifying symmetries.

The mathematical framework is established in section 3.2. It is shown that a single symmetry is sufficient to unskew the affine image, to give a one-parameter family of symmetric shapes that could have given rise to the image. A second co-planar symmetry is sufficient, in general, to uniquely determine the aspect ratio of the plane. This determines the back-projection up to a similarity transformation. This is achieved without any knowledge of the intrinsic camera parameters, an important consideration in practice since the automatic calibration of camera intrinsic parameters is a poorly conditioned, nonlinear problem. If, however, the camera aspect ratio is known, the back-projection determines slant (up to the usual two fold ambiguity often referred to as the Necker ambiguity) and tilt of the object plane. Finally, a test for non-co-planarity is given for two symmetric objects from a single image.

Throughout the chapter, results are shown from an implementation of the theory outlined in sections 3.2 and 3.3. Image back-projection and its applications are demonstrated in section 3.4 followed by discussion. The reader may care to pause at this point to view figures 3.13, 3.14; figures 3.15, 3.17; and figures 3.18, 3.19, which show typical results using the implemented program.

3.2 Mathematical framework

In this section we study the constraints on the transformation between two image contours if they are the projections of corresponding sides of a planar object with bilateral symmetry. Figure 3.1 illustrates the situation under consideration: the image contours γ and γ' are the images, assumed to be affine, of two corresponding sides, Γ and Γ' respectively, of a planar object with a bilateral (mirror) symmetry. A key attribute of affine transformations is that they preserve parallelism, so lines joining corresponding points in the image, for example the lines $\mathbf{aa}'$ and $\mathbf{bb}'$, are parallel. Affine transformations also preserve length ratios on parallel lines. In particular mid-points are preserved, so the imaged symmetry axis passes through the mid-points of $\mathbf{aa}'$ and $\mathbf{bb}'$.

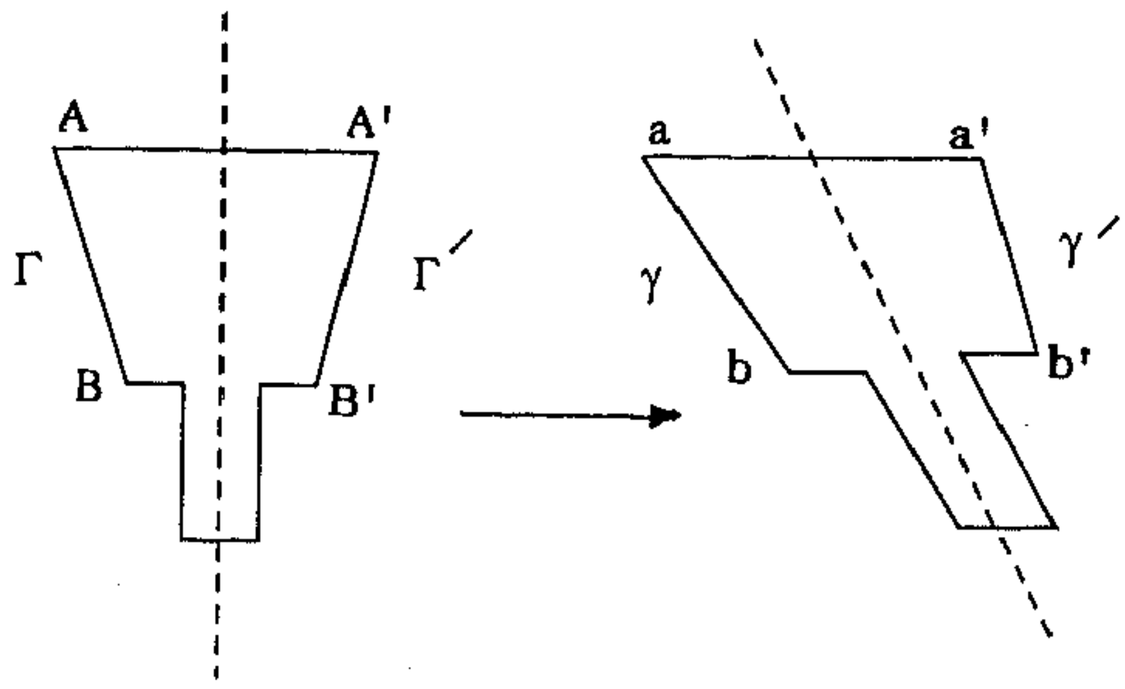

Figure 3.1: γ and γ' are the affine images of two corresponding sides of a planar object, Γ and Γ' respectively, with bilateral symmetry. Affine transformations preserve parallelism, so lines joining corresponding points in the image, for example the lines $\mathbf{aa'}$ and $\mathbf{bb'}$, are parallel. Affine transformations also preserve length ratios on parallel lines. In particular mid-points are preserved, so the imaged symmetry axis passes through the mid-points of $\mathbf{aa'}$ and $\mathbf{bb'}$.

3.2.1 Image transformation

The following theorem explores the image transformation between the affine images γ and γ' in more detail. In particular, it fixes notation that will be used subsequently to unskew images.

Theorem 1 *Suppose two curves γ and γ', as in figure 3.1, are the images of two corresponding sides of a planar object with bilateral symmetry. Suppose further that image projection can be represented by an affine transformation.*

Then the transformation between γ and γ' has the following properties:

1. *γ and γ' are related by an affine transformation. That is, if $\mathbf{x}$ is a point on γ then there is a point $\mathbf{x}'$ on γ' such that:*

$$\mathbf{x}' = \mathtt{A}\mathbf{x} + \mathbf{b} \tag{3.1}$$

where $\mathtt{A}$ is a non-singular 2×2 matrix, and $\mathbf{b}$ is a two-vector.

2. *The affine transformation $\{\mathtt{A}, \mathbf{b}\}$ satisfies the following constraints:*

(a) $\mathbf{A}^2 = \mathbf{I}$

(b) $|\mathbf{A} + \mathbf{I}| = 0$

(c) $|\mathbf{A} - \mathbf{I}| = 0$

3. *The matrix* $\mathbf{A}$ *has eigenvectors* $\mathbf{a}$ *and* $\mathbf{b}$ *(*$\mathbf{b}$ *as above) with eigenvalues* $+1$ *and* -1 *respectively. Vector* $\mathbf{a}$ *is parallel to the symmetry axis, vector* $\mathbf{b}$ *is parallel to* $\mathbf{x}' - \mathbf{x}$.

4. *The transformation has three degrees of freedom. It can be parametrised by* a, b_x, b_y *where:*

$$\mathbf{A} = \begin{bmatrix} a & -b_x(1+a)/b_y \\ -b_y(1-a)/b_x & -a \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \tag{3.2}$$

5. *With this notation, the image (skewed) symmetry line is:*

$$(1-a)b_y x + (1+a)b_x y - b_x b_y = 0. \tag{3.3}$$

The proof is given in appendix A.1. Note that the affine transformation has only three degrees of freedom (given by a, b_x, b_y in the statement of the theorem) as claimed in the introduction. The theorem also applies if the original object has affine skewed symmetry, since an affine image of an affine transformed object with bilateral symmetry has the same properties.

3.2.2 Back-projection

In this section we consider the extent to which we can unskew images such as those shown in figure 3.10. Evidently, some of the skewed objects in figure 3.10 have only a single symmetry, in which case the best one can hope for in general (that is, without mobilising further knowledge) is to unskew the image to a single parameter family of symmetric shapes that corresponds to tilting the symmetric shape backwards while preserving the direction of the symmetry axis. If, on the other hand, there is more than one symmetry axis, then one might hope to combine the information from two or more such to uniquely unskew the image. This insight is proved in the following theorem.

Theorem 2 *Suppose we have an (un-calibrated) image of one or more co-planar symmetric objects then*

1. **One symmetry**
   If there is only one symmetry present, the image can be back-projected, modulo a similarity, to form a one-parameter family of symmetric shapes that could have given rise to the image.

2. **Two symmetries**
   In the case that two symmetries are present in the image, the image can be back-projected uniquely, modulo a similarity, provided that the two symmetry axes are neither parallel nor orthogonal (either in the image or in space, since the camera is assumed affine).

Proof

First some notation for back-projections. Suppose the affine transformation relating the object and image planes is given by:

$$\mathbf{X} = \mathtt{U}\mathbf{x} + \mathbf{B} \tag{3.4}$$

where $\mathbf{x}$ is the 2D image vector, $\mathbf{X}$ is the corresponding two-vector in back-projected planar scene, $\mathbf{B}$ is a two-vector translation and $\mathtt{U}$ is the 2×2 linear transformation matrix with *det* $\mathtt{U} \geq 0$ responsible for back-projection. It can be shown [7] that the transformation $\mathtt{U}$ can be decomposed as:

$$\mathtt{U} = \lambda_1 \mathtt{R}(\theta)\mathtt{P}(\lambda, \tau) \tag{3.5}$$

where $\mathtt{P}(\lambda, \tau)$ is a symmetric matrix:

$$\mathtt{P}(\lambda, \tau) = \mathtt{R}(\tau) \begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} \mathtt{R}(-\tau) \tag{3.6}$$

with

$$\mathtt{R}(\tau) = \begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix}.$$

This particular decomposition of U makes explicit its four degrees of freedom in a way that corresponds to the processes of projection: the linear transformation U consists of an isotropic scaling by λ_1, a rotation about the optic axis by $\mathtt{R}(\theta)$, and an expansion by λ in the direction $\boldsymbol{\tau} = (\cos\tau, \sin\tau)$ (the eigenvector of $\mathtt{P}(\lambda, \tau)$).

The eigenvectors of A (refer to theorem 1), say a and b, back-project to vectors respectively parallel to and orthogonal to the symmetry axis. In the object plane, therefore, the scalar product (Ua).(Ub) = 0 and consequently:

$$\mathbf{a}^t\mathtt{U}^t\mathtt{U}\mathbf{b} = 0. \tag{3.7}$$

The matrix $\mathtt{V} = \mathtt{U}^t\mathtt{U}$ is clearly symmetric and it is positive definite. Let the components of V be given by:

$$\mathtt{V} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}, \tag{3.8}$$

then we have from equation (3.7)

$$\begin{pmatrix} a_x b_x & a_x b_y + a_y b_x & a_y b_y \end{pmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = 0 \tag{3.9}$$

This is a linear constraint on α, β, γ. Two such constraints determine the *ratio* $\alpha : \beta : \gamma$. The sign is fixed by the requirement that V is positive definite, so that trace $\mathtt{V} = \alpha + \gamma > 0$. This is sufficient to determine λ and τ as is shown in the following lemma.

Lemma: *The ratio $\alpha : \beta : \gamma$, with sign chosen so that $\alpha + \gamma > 0$, determines λ up to a four fold ambiguity, and τ up to a two fold ambiguity.*

Note that λ^2 is the ratio of the eigenvalues of V, and τ is the rotation angle to the eigen-directions. From equations (3.5) and (3.6)

$$\mathbf{V} = \lambda_1^2 \mathrm{P}(\lambda^2, \tau) \tag{3.10}$$

The trace and determinant of V give two equations for λ^2

$$\begin{aligned} \text{trace } \mathbf{V} &= \alpha + \gamma = \lambda_1^2(1+\lambda^2) \\ \det \mathbf{V} &= \alpha\gamma - \beta^2 = \lambda_1^4 \lambda^2 \end{aligned}$$

Eliminating λ_1^2,

$$\frac{(\text{trace } \mathbf{V})^2}{\det \mathbf{V}} = 4\mu \tag{3.11}$$

$$\frac{(1+\lambda^2)^2}{\lambda^2} = \frac{(\alpha+\gamma)^2}{\alpha\gamma - \beta^2} = 4\mu \tag{3.12}$$

Solving this gives $\lambda^2 = \{\lambda'^2, 1/\lambda'^2\}$ where $\lambda'^2 = 2\mu - 1 + 2\sqrt{\mu(\mu-1)}$. Or, equivalently $\lambda = \pm\sqrt{\mu} \pm \sqrt{\mu - 1}$, which are four solutions of the form $\{\lambda', 1/\lambda', -\lambda', -1/\lambda'\}$ with $\lambda' = \sqrt{\mu} + \sqrt{\mu - 1}$.

The rotation angle, τ is obtained from equations (3.6) and (3.10)

$$\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} = \lambda_1^2 \mathrm{R}(\tau) \begin{bmatrix} \lambda^2 & 0 \\ 0 & 1 \end{bmatrix} \mathrm{R}(-\tau) \tag{3.13}$$

Rearranging gives

$$\begin{bmatrix} \cos\tau & \sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix} \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix} = \lambda_1^2 \begin{bmatrix} \lambda^2 & 0 \\ 0 & 1 \end{bmatrix}$$

From the off diagonal elements we obtain:

$$-\beta \sin^2\tau - (\alpha - \gamma)\cos\tau \sin\tau + \beta \cos^2\tau = 0$$

from which

$$\tan 2\tau = \frac{2\beta}{\alpha - \gamma} \tag{3.14}$$

This gives four solutions for τ. If $\tau^* = (1/2)\tan^{-1}[(2\beta)/(\alpha-\gamma)]$ then the four solutions are $\{\tau^* + n\pi/2\}$ for $n = 0, 1, 2, 3$. Two of the solutions are simply due to a clockwise rotation as opposed to counter-clockwise, so may be disregarded. It is only necessary to determine if τ is in the first or second quadrant. From (3.13)

$$\beta = \lambda_1^2(\lambda^2 - 1)\cos\tau\sin\tau = (1/2)\lambda_1^2(\lambda^2 - 1)\sin 2\tau$$

so that $\text{Sign}(\beta/(\lambda^2-1)) = \text{Sign}(\sin 2\tau)$ and if $\text{Sign}(\beta/(\lambda^2-1)) \geq 0$ then $0 \leq \tau \leq \pi/2$, otherwise $\pi/2 < \tau \leq \pi$. Consequently, there are two solutions for τ corresponding to the two solutions, $\{\lambda'^2, 1/\lambda'^2\}$ above for λ^2. □

This proves the lemma, now we can return to the proof of the theorem. In the following we take $\lambda = \sqrt{\mu} + \sqrt{\mu - 1}$. The other solutions differ only by similarity transformations. If $\lambda = \sqrt{\mu} + \sqrt{\mu - 1}$ then for real solutions $\mu \geq 1$ and consequently $\lambda \geq 1$. Hence $\text{Sign}(\beta) = \text{Sign}(\sin 2\tau)$ and this uniquely determines τ.

1. **One symmetry**
   Equation (3.9) has a one parameter family of solutions for the ratio $\alpha : \beta : \gamma$. Correspondingly there is a one parameter family of solutions for λ and τ.

2. **Two symmetries**
   Two symmetries generate two constraint equations (3.9):

$$\mathrm{M}\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = 0$$

where

$$\mathrm{M} = \begin{bmatrix} a_x^1 b_z^1 & a_x^1 b_y^1 + a_y^1 b_x^1 & a_y^1 b_y^1 \\ a_x^2 b_z^2 & a_x^2 b_y^2 + a_y^2 b_z^2 & a_y^2 b_y^2 \end{bmatrix} \tag{3.15}$$

Provided the matrix M is of rank 2 this uniquely determines the ratio $\{\alpha : \beta : \gamma\}$ (and consequently, from the lemma, λ and τ). It can be shown that M drops rank if any of the vectors $\{\mathbf{a}^1, \mathbf{b}^1, \mathbf{a}^2, \mathbf{b}^2\}$ are parallel, hence the clause in the theorem. The optimal solution when there are more than two constraints is discussed in section 3.4. □

3.3 Detecting symmetries

As shown in section 3.2, corresponding sides, Γ and Γ' of a symmetric planar object project to image curves γ and γ' respectively. Even though the projection ($\Gamma \rightarrow \gamma$ and $\Gamma' \rightarrow \gamma'$) is by a general affine transformation, the image curves γ and γ' are related by a three dimensional subset of the (6 dof[1]) planar affine group. In this section we describe how these results can be utilised to detect efficiently such image pairs.

Intra-image curve matching has much in common with the inter-image curve matching necessary for model based vision, and approaches developed for that area can be used to advantage here. In particular, and this introduces the second theme of invariants foreshadowed in the introduction, the use of invariants as index functions avoids the cost of a six-dimensional search over transformation parameters [39,64]. The three stages of an implemented algorithm are described in the following sections.

3.3.1 Generating and matching affine invariants

Two curves that are related by an affine transformation have the same affine invariants. The converse is not necessarily true, but invariants can usefully be used to generate hypotheses for matching, which can subsequently be tested. Briefly, a function $I(\Gamma)$ of a curve Γ is an *invariant* if $I(\gamma) = |\mathtt{U}|^w I(\Gamma)$ where γ is the image of Γ (refer to figure 3.1)under the affine transformation as defined in equation (3.4). The exponent w is the weight of the invariant. If $w = 0$ then the invariant is absolute, otherwise it is relative. Note that in order to determine local symmetries, the invariant cannot depend on global properties of the curve. Examples of (semi-local) affine invariants for smooth curves are given below.

[1]degrees of freedom

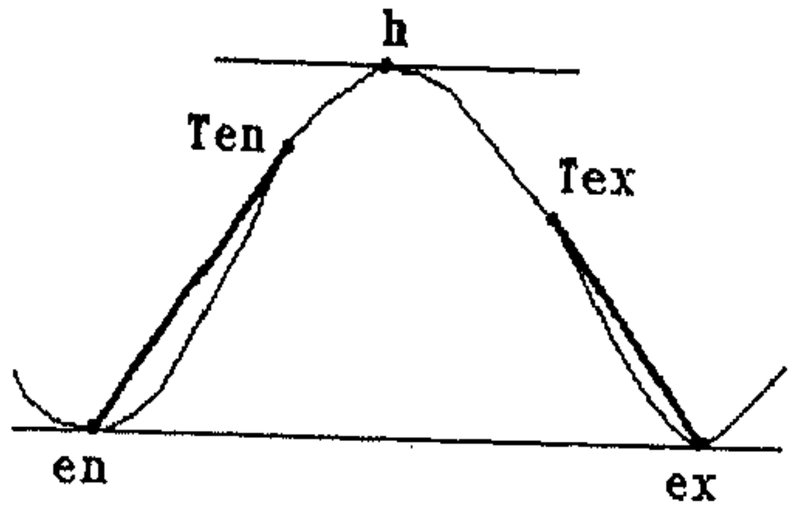

Figure 3.2: Examples of distinguished points for a non-convex curve under affine transformations. Points *en* and *ex* mark the entrance and exit of the concavity determined by the bi-tangent line. Further distinguished points are constructed from these points: *Ten* is the point on the curve which is tangent to a ray based at *en* (similarly for *Tex* and *ex*); *h* is determined by the line parallel and furthest from the bi-tangent. Apart from *h* these distinguished points are also preserved by projective transformations. Examples of these points on an image curve are shown in figure 3.6.

Unlike model based vision, where *absolute* invariants are needed, *relative* invariants suffice in this case. To see this, consider two symmetry-related curves in the object plane. These are related by an affine transformation with determinant -1 (since the transformation is a reflection). Consequently, affine invariants of each side of the shape are equal modulo a sign. In the image, invariants are multiplied by $|U|^w$, which is unknown, but which is the same for both sides. Thus, relative affine invariants of each side have the same magnitude.

Matching on invariants can be implemented as an $O(n)$ complexity process by the use of hashing (where n is the number of curves) [62,64]. We have implemented the simpler $O(n^2)$ algorithm, since n is small in the cases we have experimented with. It is straightforward to implement the more complicated algorithm.

Affine semi-local invariants

For a non-convex curve, following Lamdan *et al.* [39], we exploit concavities, by constructing a bi-tangent across the concavity and determining the interior point on the concavity curve with tangent parallel to the bi-tangent. See figure 3.2. This generates three distinguished points. This particular choice of points has the

Table 3.1: Affine invariant values for objects in figure 3.3. M_x and M_y are moments about x and y axes respectively.

Invariant	Spanner		Spoon		Hex Spanner		Plier	
Area (Image space)	1064	1056	3694	3855	2719	2723	4398	3814
M_x (Canonical frame)	10.85	11.38	32.13	31.44	18.61	17.13	2.57	2.03
M_y (Canonical frame)	102.15	99.71	86.17	84.41	92.72	98.88	82.59	78.82

advantage that it does not depend globally on the curve. Consequently, if part of the curve is occluded or missed because of segmentation problems, *local* symmetries can still be detected. Affine invariants are generated from the concavity curve:

1. **Area in the image space**
   This is a relative invariant. The area used is that of the triangle defined by the three concavity distinguished points (en, ex and h).

2. **Moments in the canonical frame**
   Significant concavities are mapped to a canonical frame [39] consisting of an equilateral triangle with vertices at $(-1,0)$, $(1,0)$ and $(0,\sqrt{3})$ by using the affine basis triplet points of the concavities. The x and y moments of the concavity in the canonical frame are used as invariant indexes.

Table 3.1 lists the invariant values for the objects in figure 3.3. These differ, in general, by less than 2% for symmetry related concavities. Figure 3.4 shows the matched concavity pair extracted from figure 3.3. For the plier, invariant values are not consistent because of the thickness of the handles. The handles cause two problems: firstly, they are rounded so (as in the case of an extremal boundary) the surface curves projecting to the outline will be space curves and not mirror pairs in general; secondly, and more important in this case, the handles and jaw are not in the same plane.

For a convex curve segment distinguished points can be obtained from the anti-symmetry set [8].

3.3.2 Determining the affine transformation

Having found two curves with matching invariant(s), the next stage is to determine if the curves are affine related. This is achieved by extracting a number of *dis-*

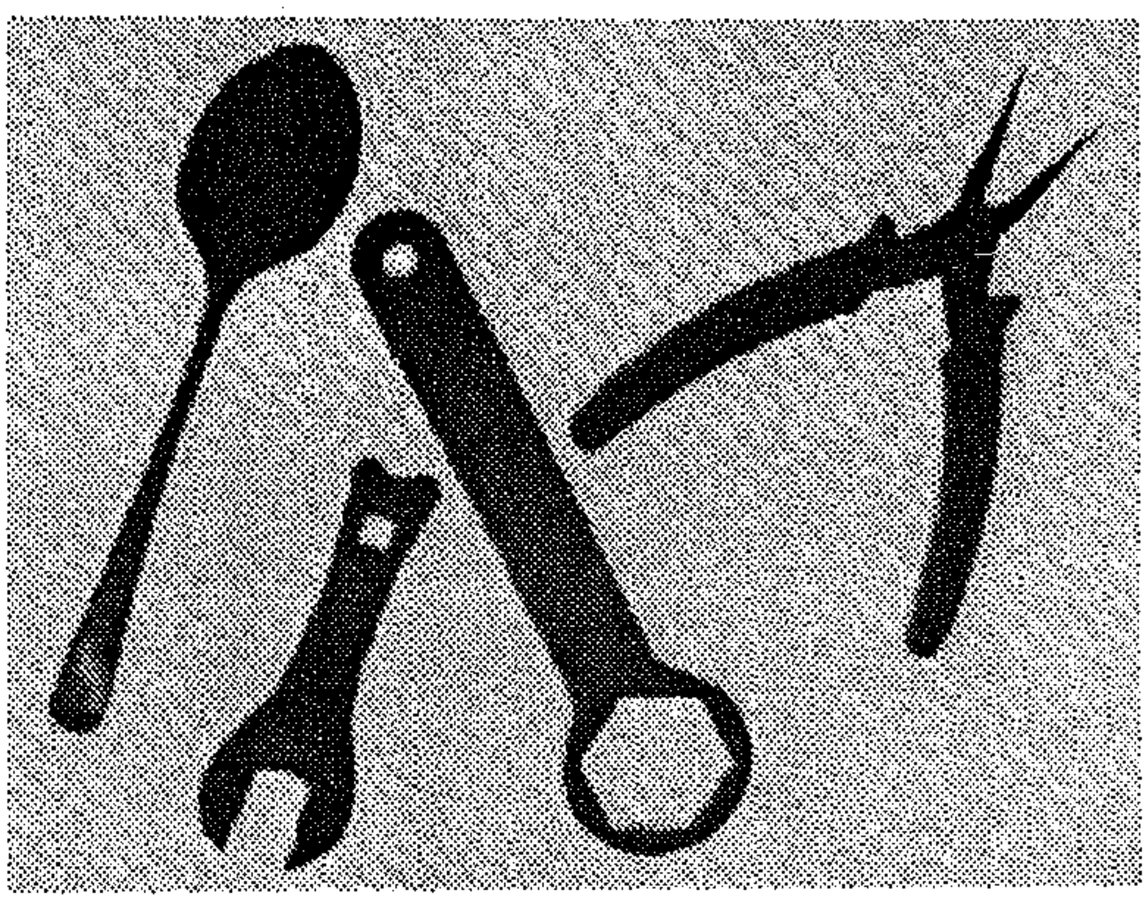

Figure 3.3: Affine image of four objects with bilateral symmetries.

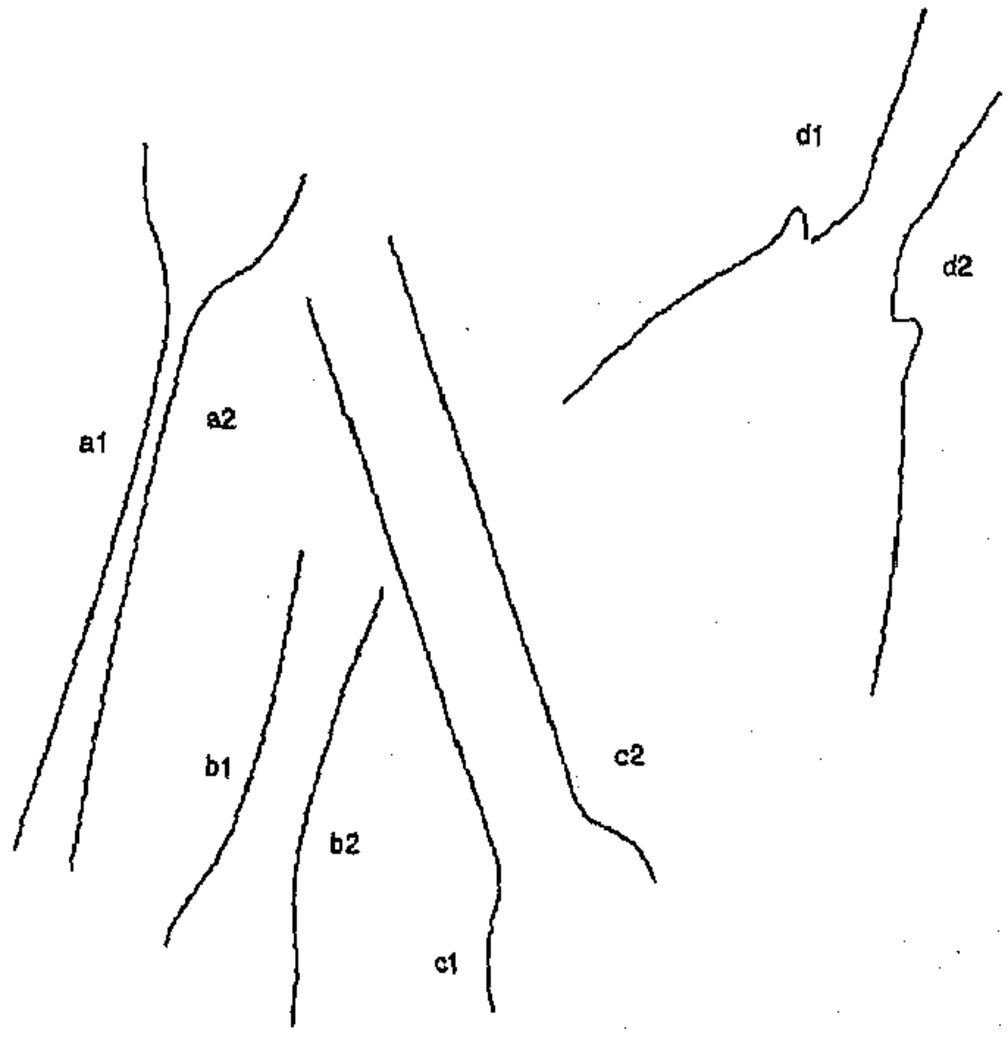

Figure 3.4: Matched concavities (a1, a2), (b1, b2), (c1, c2) and (d1, d2) are extracted from the symmetric objects of figure 3.3.

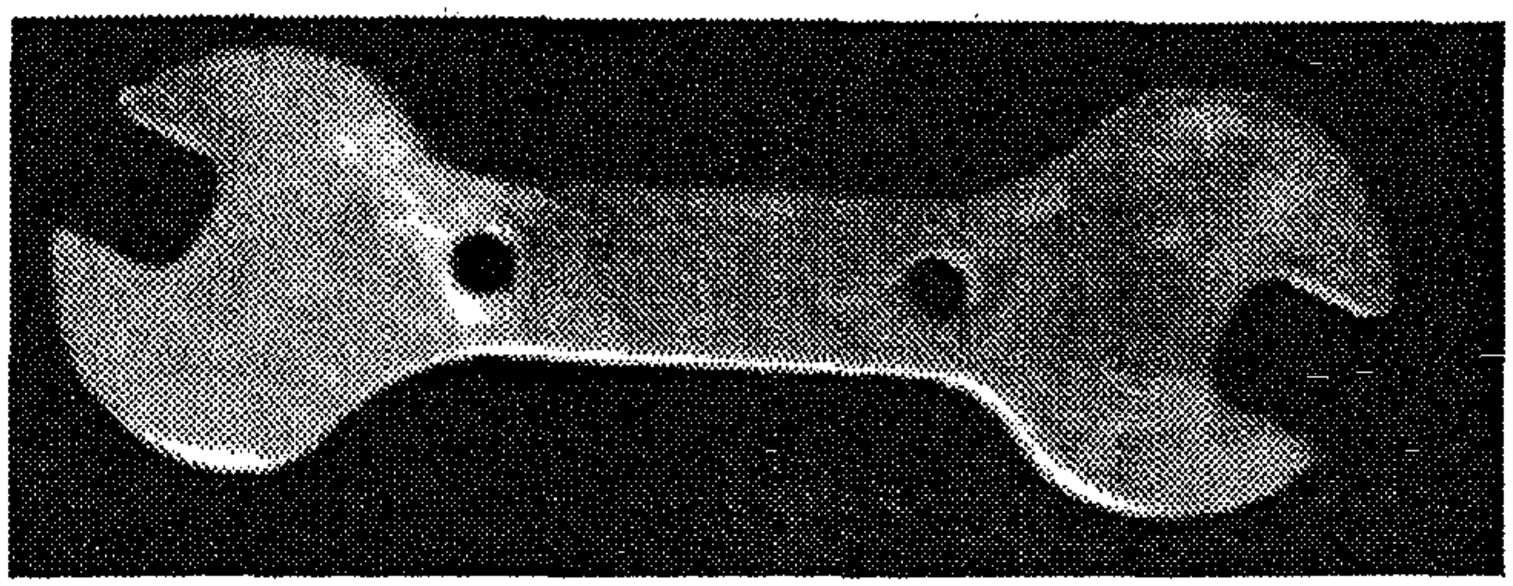

Figure 3.5: Affine view of spanner.

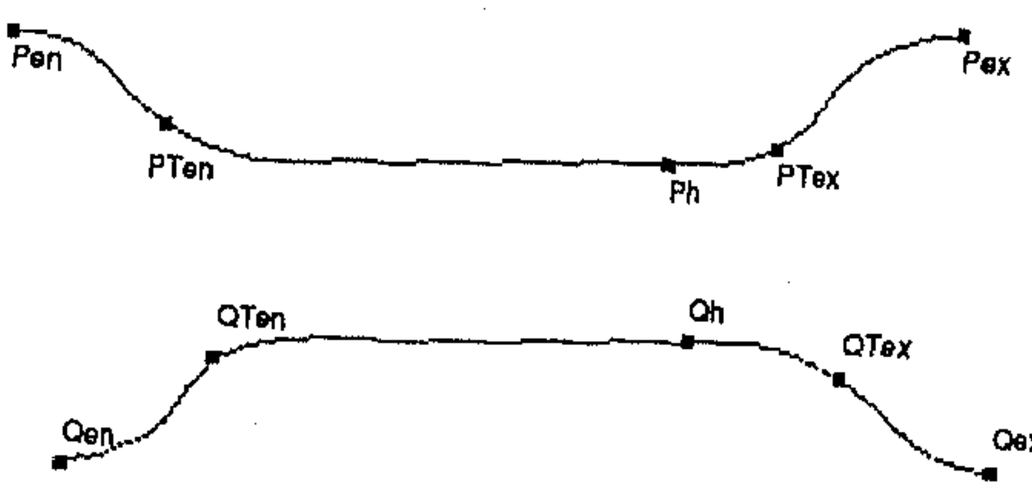

Figure 3.6: Distinguished points for edge curves extracted from the image of figure 3.5. The notation of points for the curves P and Q is defined in figure 3.2.

tinguished points on each curve, and determining an affine transformation between these point sets. Three points are required to determine a general affine transformation. Distinguished points are curve "markers" that can be determined before and after a transformation. A number of examples are shown in figure 3.2. They include points preserved by projectivities (such as inflections, bi-tangent contact points, "cast" tangents) as well as those exclusively preserved by affinities (such as points defined by parallel lines). Examples are shown in figure 3.6. Note, points are ordered by the curve so the correspondence problem is greatly simplified.

3.3.3 Verifying subset membership

As noted in section 3.2, the image curves are related not by a general affine transformation (with six degrees of freedom), but by a three parameter subspace. If the affine transformation does not lie in this subspace, then the two curves cannot be symmetry related and as a consequence, inter-curve reflections, as in figure 3.9, will not work. Note that if two curves *are* symmetry related then two points are sufficient to determine the transformation of equation (3.2). When more points are available the form of the transformation is used as a constraint (via a Lagrange multiplier) in a least squares estimator. Details are given in the next section. If the curves are affine related then one side can be superimposed on the other. An example is shown in figure 3.9.

3.3.4 Implementation and results

Feature extraction Image contours are extracted using a local implementation of the Canny [15] edge detector. Significant concavities are extracted for each closed contour after computing a convex hull and setting a threshold on concavity height and width. Bi-tangents are found via a dual space construction [62] and this determines the concavity entrance (en) and exit points (ex). Concavity height point (h) is determined by the line parallel and furthest from the bi-tangent.

Concavity matching For each closed contour in the scene, matched concavities are detected using affine invariant indexes as described in section 3.3.1. Examples of the three points, en, ex and h, used as an affine basis are shown in figure 3.8. Corresponding points in the matched concavity pair are determined from the tracing order (clockwise or anti-clockwise) of the image contour from which concavities are extracted. These point correspondences are used to determine the affine transformation between corresponding curves.

Affine transform The next step is to determine if the affine transformation arises from a reflectional symmetry of the object curves i.e. whether it lies in the 3 dof subspace defined by the constraints of equation (3.2). In practice the affine transformation determined from the three affine basis points is not sufficiently accurate, so extra correspondences are included via a pseudo inverse. Two additional points,

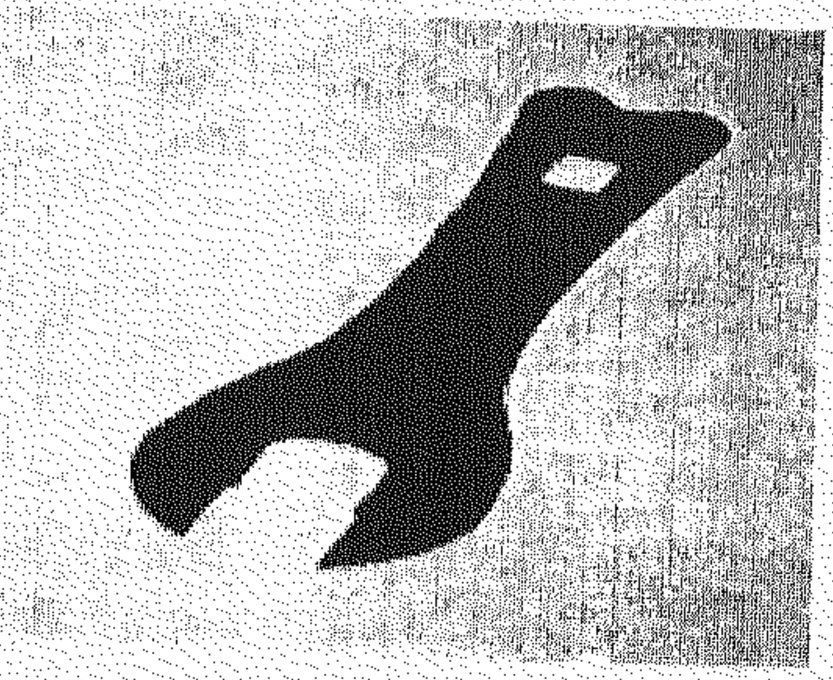

Figure 3.7: Affine view of symmetric spanner.

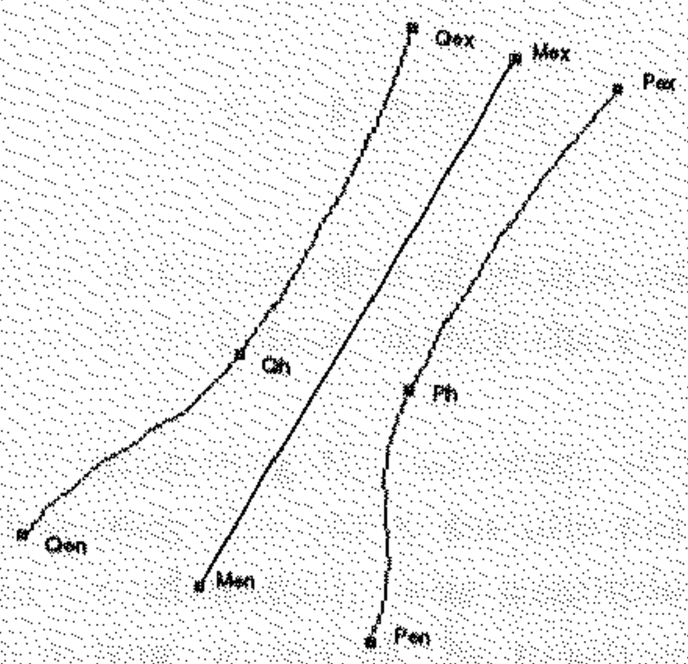

Figure 3.8: Matched concavities extracted from symmetric spanner of figure 3.7. A quick, though not as accurate, method for determining the affine symmetry line is to join the mid-points M_{en} and M_{ex} (which are preserved by affinities) of corresponding distinguished point sets.

Table 3.2: As the accuracy of the affine transformation is improved, it more closely satisfies the constraints of equation (3.2). The table gives the computed A and b elements when they are calculated either (a) directly from three points, or (b) via a pseudoinverse from five points. The transformation is for the points obtained from figure 3.20.

No of Pts	Affine elements solving eqn. (3.1) and (3.17)						Elements calculated as eqn. (3.2)	
	a_{11}	a_{12}	b_x	a_{21}	a_{22}	b_y	$a_{12} = -b_x(1+a)/b_y$	$a_{21} = -b_y(1-a)/b_x$
3	0.89	0.68	-72.16	0.49	-0.79	158.57	0.86	0.24
5	0.77	0.97	-87.79	0.48	-0.75	157.15	0.98	0.41

marked T_{en}, T_{ex} in figures 3.2 and 3.6, are the points of tangency to the extracted curve drawn through the cavity entrance and exit points (these are determined from the convex hull). In straightforward notation, equation (3.1) is rewritten as

$$\mathtt{P}\mathbf{X} = \mathtt{Q} \tag{3.16}$$

where $\mathbf{X}$ is a six-vector formed from the elements of A and b. This is solved using a pseudo-inverse as

$$\mathbf{X} = \mathtt{P}^t(\mathtt{P}\mathtt{P}^t)^{-1}\mathtt{Q} \tag{3.17}$$

Table 3.2 demonstrates that as the number of points increases the accurately determined affine transformation does indeed satisfy the constraints of equation (3.2).

An alternative method for improving the accuracy of the affine transformation is to minimise differences between the curve on one side and the other side affine transformed (so it should be identical). For example differences of area, or the distance between corresponding points of the matched curve could be used as a measure. This has not been implemented, but the accuracy of the affine transformation computed from the pseudo-inverse is demonstrated in figure 3.9 (edge image of figure 3.20) where one side Q is "reflected" onto the other side P.

Affine symmetry axis Having determined the affine transformation, the symmetry axis is given by equation (3.3). Since mid-points are preserved by affine transformations, the mid-points of lines joining corresponding distinguished points lie on the symmetry axis. This provides a quick, though not as accurate, method for determining symmetry axis. An example is shown in figure 3.8 where the line

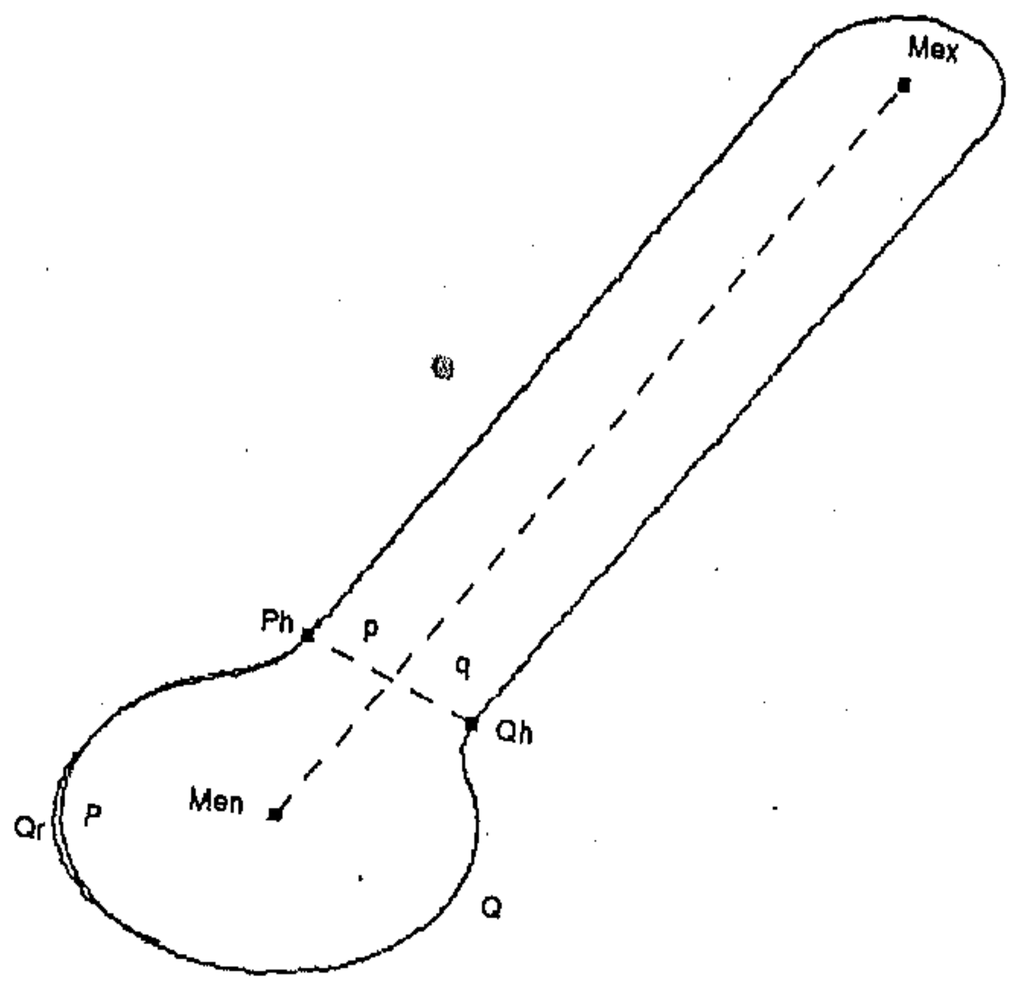

Figure 3.9: Q_r is the image "reflection" of Q *onto* P *using an affine transformation* computed from five point correspondences for the hex spanner shown in figure 3.20.

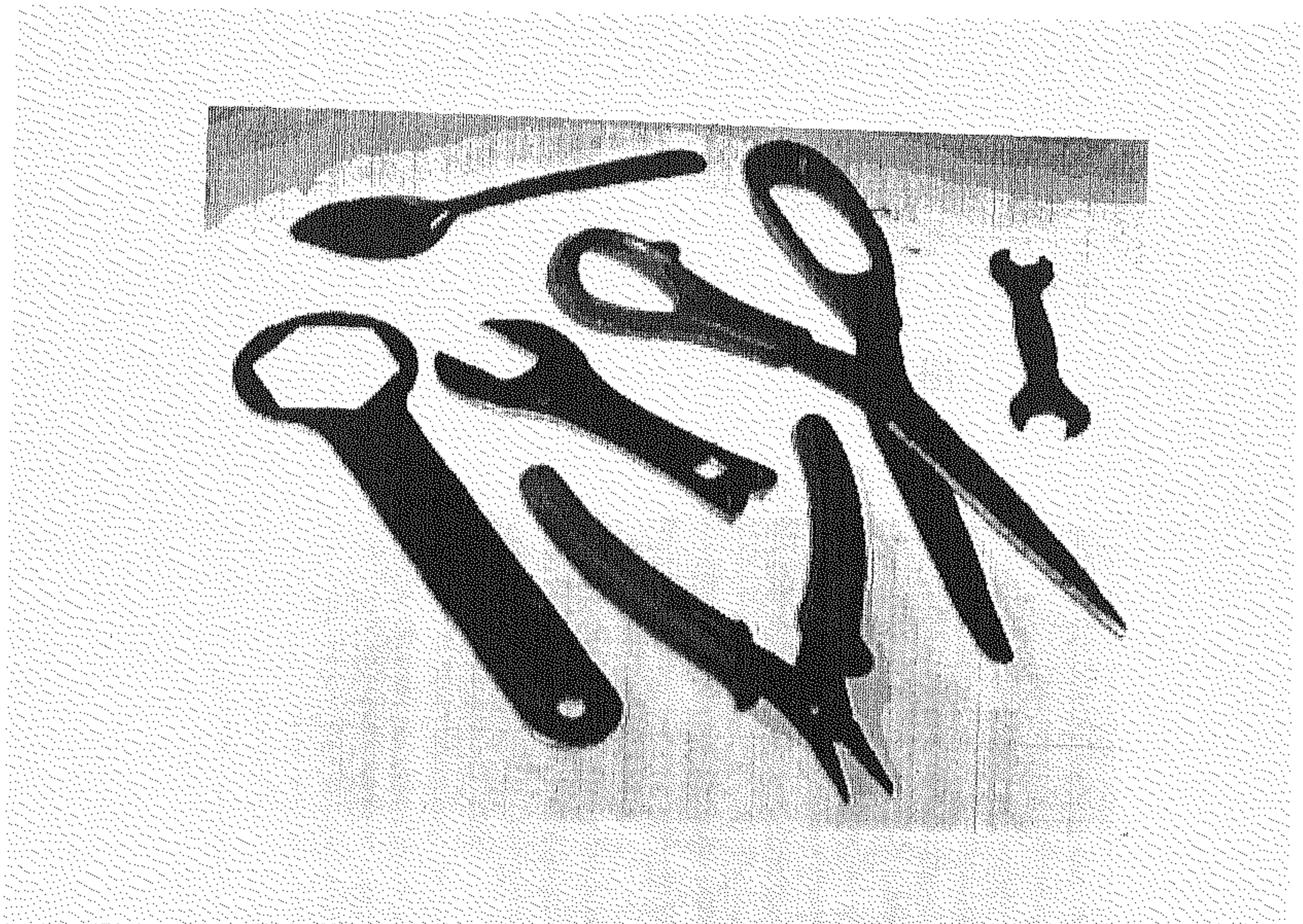

Figure 3.10: Affine scene of globally, partially and non-symmetric objects.

joining mid-points M_{en} and M_{ex} of corresponding distinguished point sets gives the affine symmetry line.

3.3.5 Global symmetries

Clearly, a local symmetry between two concavities does not imply a global reflectional symmetry for the whole object. For example, while the symmetric spanner shown in figure 3.7 has a global reflectional symmetry, the spanner shown in figure 3.5 does not. To test for global symmetry, the local symmetry line is extended in both directions whilst there is evidence that a symmetry with this axis exists. This is the case if for each point on one side of symmetry line there is a corresponding point on the other side in the direction of b (as defined in equation (3.2)) and at the same distance from the symmetry line.

Figure 3.10 is an affine scene containing globally, partially symmetric and non-symmetric objects. Globally symmetric contours are correctly determined as shown in figure 3.11.

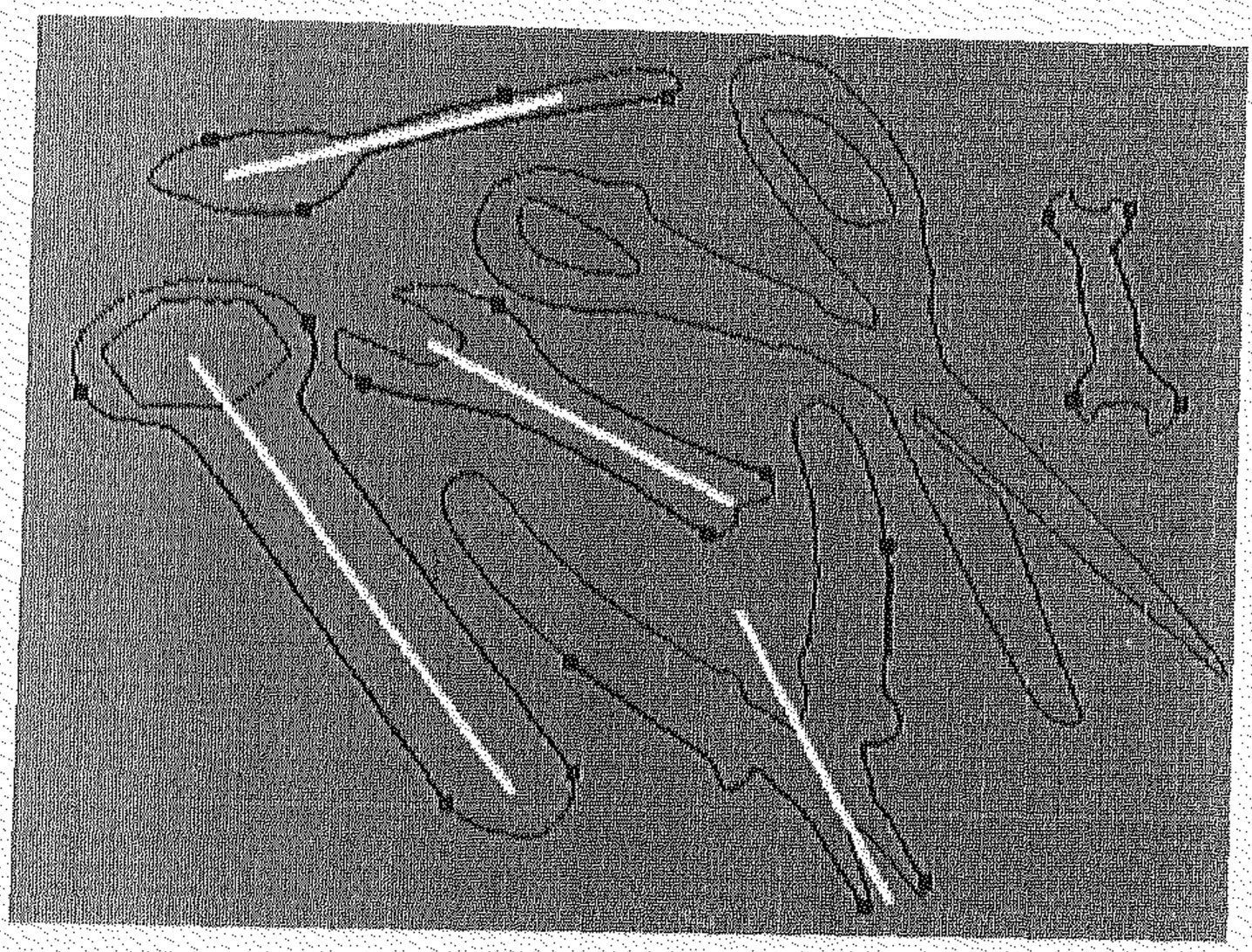

Figure 3.11: Outline curves and concavity entrance and exit points for the objects in figure 3.10. Symmetry lines are only drawn where the object is determined to be globally symmetric.

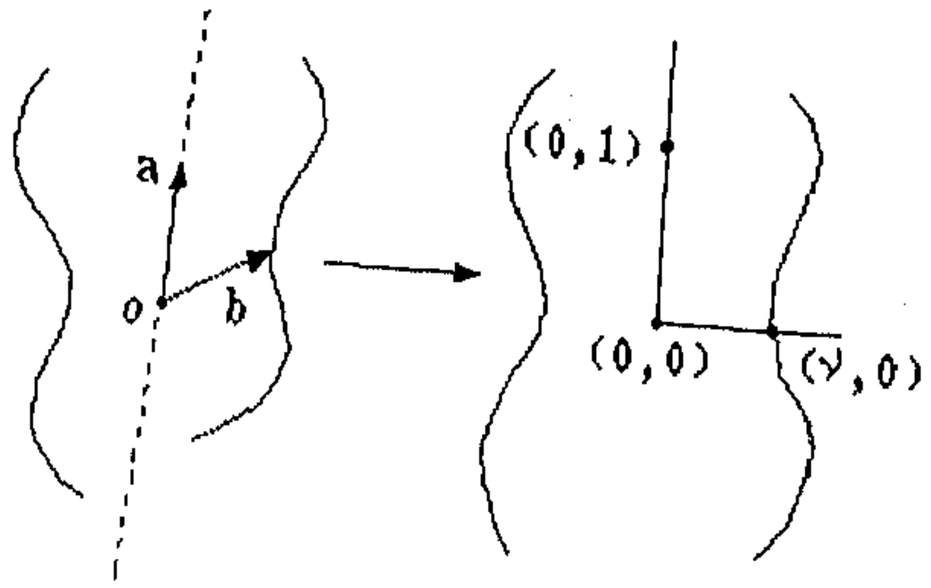

Figure 3.12: Mapping of guiding vectors **a** and **b** to *unskewing frame* with vertices at $(0,1)$, $(0,0)$ and $(\nu,0)$.

3.4 Applications

3.4.1 Back-projection

Here we determine the back-projection to the object plane using the results of section 3.2. Note back-projection does not require camera aspect ratio (or any of the intrinsic parameters). We first give an intuitive and simple construction for determining the back-projection and its uniqueness, which is applicable for two or less symmetries.

Consider an image consisting of two co-planar objects with single bilateral symmetries. Determine the skewed symmetry axes of each object (say by joining the mid-points of corresponding distinguished points), and choose an origin on one of the symmetry axes, with vector **a** on the axis, and **b** parallel to the lines joining corresponding points. Henceforth, the vectors **a** and **b** are called guiding vectors. See figure 3.12. The back-projection is achieved in two stages.

1. Unskew the first object by determining the transformation that maps **a** and **b** to the points $(0,1)$ and $(\nu,0)$ (*unskewing frame*). This determines three of the degrees of freedom of **U** including the arbitrary rotation and isotropic scaling, but does not determine the object plane aspect ratio. Explicitly

$$\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} a_x^1 & b_x^1 \\ a_y^1 & b_y^1 \end{bmatrix} = \begin{bmatrix} 0 & \nu \\ 1 & 0 \end{bmatrix}$$

So,

$$\mathtt{U} = \begin{bmatrix} 0 & \nu \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_x^1 & b_x^1 \\ a_y^1 & b_y^1 \end{bmatrix}^{-1}$$

yielding the expected one parameter (ν) family of solutions.

2. Now, ν is determined by enforcing that the second object should also be unskewed. We have:

$$\mathbf{a}^{2t}\mathtt{U}^t\mathtt{U}\mathbf{b}^2 = 0 \tag{3.18}$$

Multiplying out gives:

$$\mathbf{a}^{2t} \begin{bmatrix} a_x^1 & b_x^1 \\ a_y^1 & b_y^1 \end{bmatrix}^{-t} \begin{bmatrix} 1 & 0 \\ 0 & \nu^2 \end{bmatrix} \begin{bmatrix} a_x^1 & b_x^1 \\ a_y^1 & b_y^1 \end{bmatrix}^{-1} \mathbf{b}^2 = 0 \tag{3.19}$$

which is a linear equation for ν^2. Note that if $\mathbf{b}^1$ is parallel to $\mathbf{b}^2$ (and consequently $\mathbf{a}^1$ is parallel to $\mathbf{a}^2$) then the quadratic form in equation (3.19) is identically zero and there is no constraint on ν. Similarly, there is no constraint if $\mathbf{a}^1$ is parallel to $\mathbf{b}^2$ (and consequently $\mathbf{b}^1$ is parallel to $\mathbf{a}^2$). This occurs if the symmetry axes of both objects are parallel or orthogonal. In this case both objects are unskewed by the first stage.

This formulation is, of course, equivalent to section 3.2.2, and either can be used if there are two symmetries present. If there are more than two symmetries, where a least squared solution is required, then the above method is not easily generalisable. However, the formulation of section 3.2.2 is not restricted. Its application in a least-squared solution is described below.

Figure 3.13 and 3.14 show examples of object pairs before and after back-projection. The angle between the vectors $\mathbf{a}^i$ and $\mathbf{b}^i$ before and after are given in table 3.3. Accurate back-projection requires accurate determination of these vectors (which are

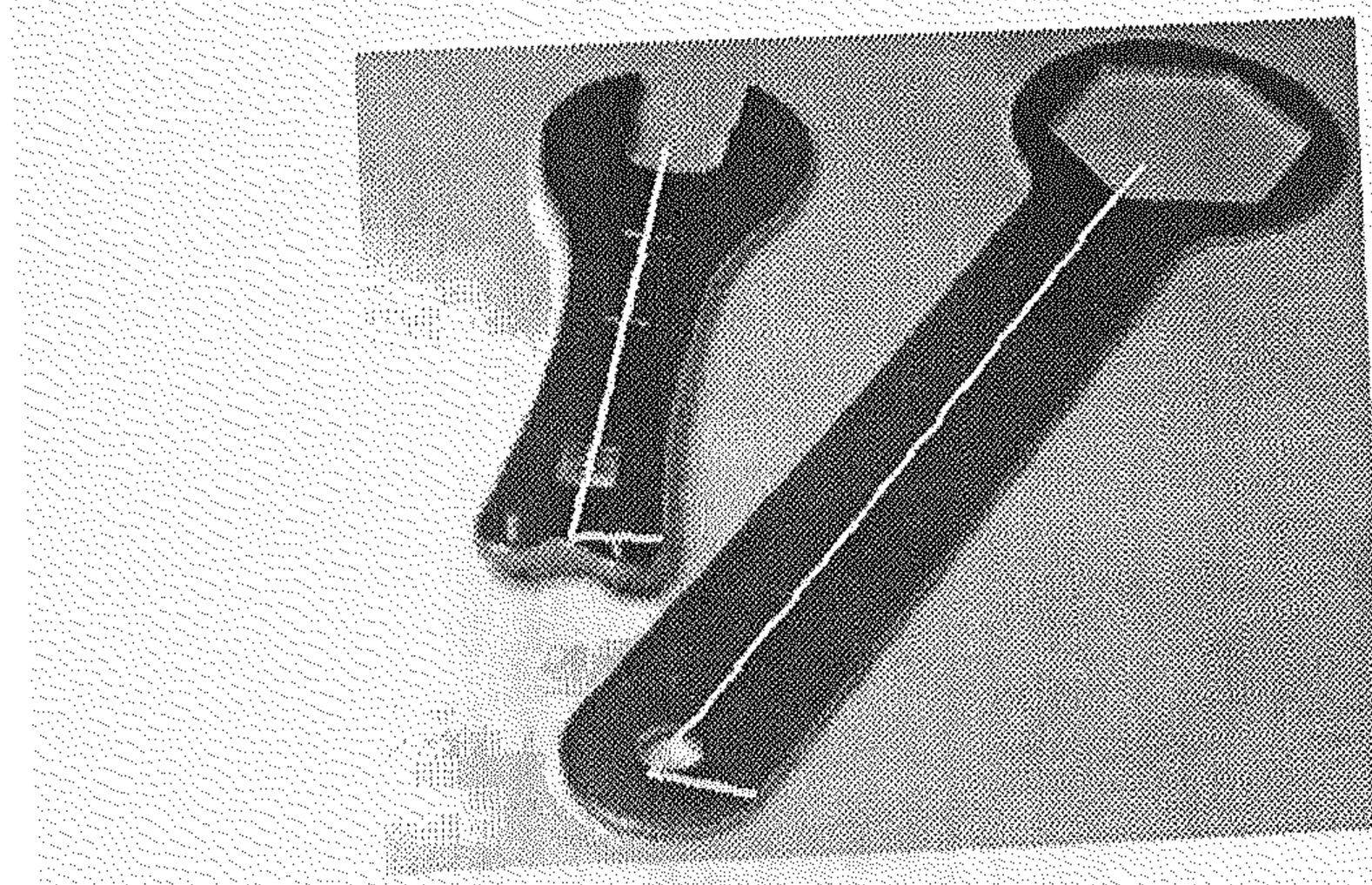

Figure 3.13: Affine view of two objects with superimposed guiding vectors. The vectors are clearly not perpendicular.

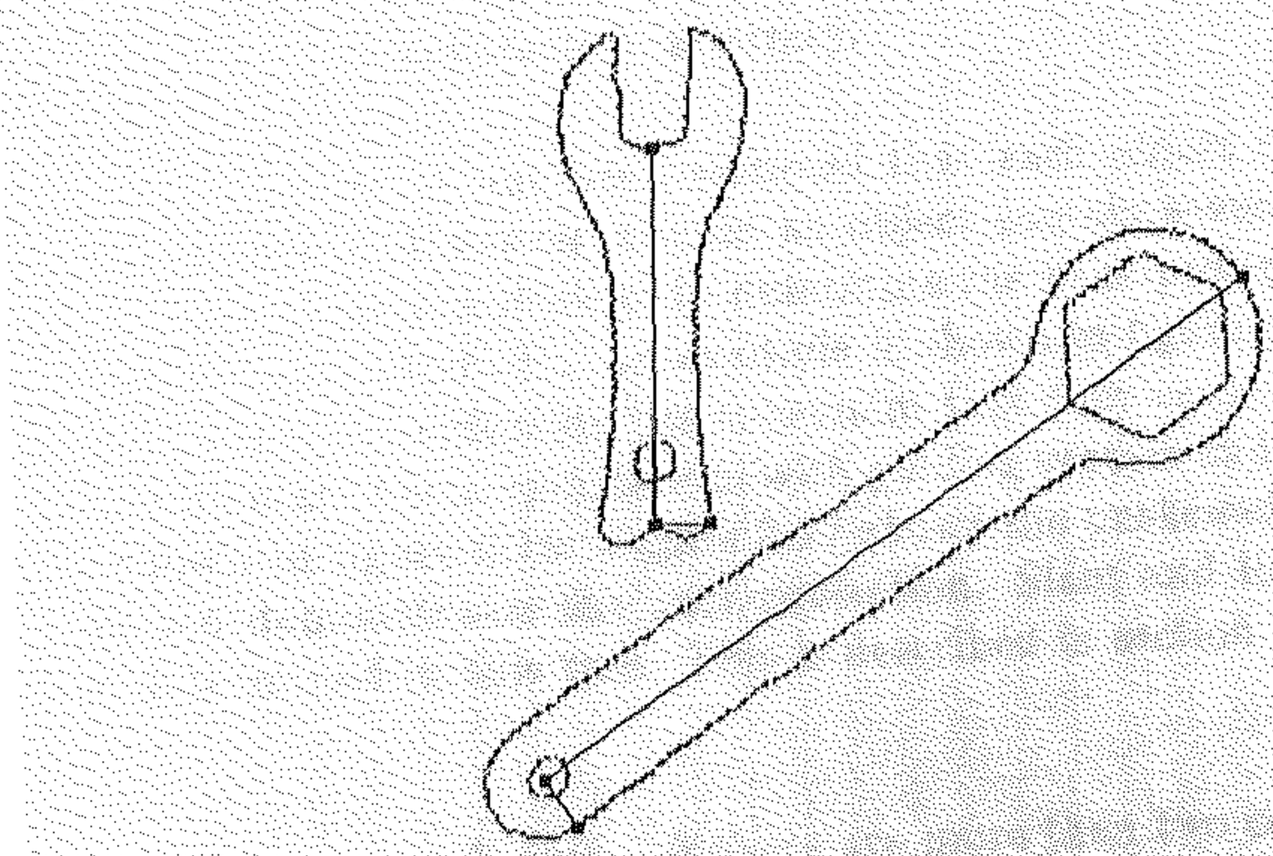

Figure 3.14: Unskewed image of the spanner and hex spanner of figure 3.13.

Table 3.3: Angles between guiding point triplet before and after unskewing for objects in figure 3.13.

Objects	Spanner	Hex spanner
Initial Angle	76.4	81.9
Final Angle	90.0	90.0

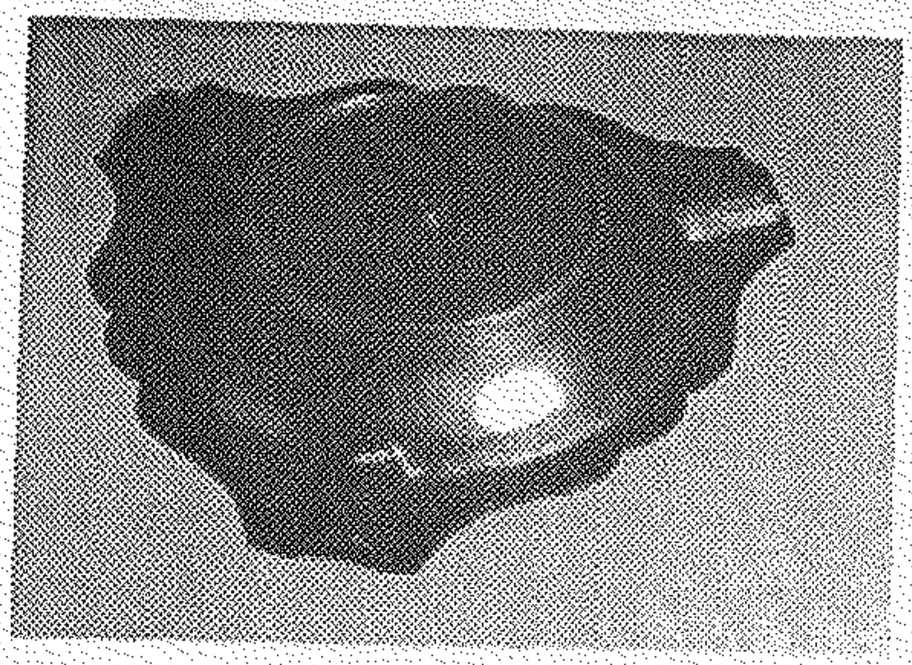

Figure 3.15: Affine image of an object with multiple local symmetries.

the eigenvectors of the matrix A). In practice we find that five point correspondences and use of a pseudo-inverse, as described in section 3.3.4, are sufficient to determine A to a satisfactory accuracy.

Up to this point it has been assumed that each object only contributes a single local symmetry. However, should an object contain several local symmetries then this object alone is sufficient to determine the back-projection (provided the usual conditions are satisfied). An example is shown in figure 3.15. We have taken two local symmetries enforcing the constraint that they cannot be mutually parallel or perpendicular. Local symmetries with guiding vectors and the matching concavities are shown in figure 3.16. Figure 3.17 shows the unskewed image of the object and table 3.4 demonstrates the back-projection. Slant and tilt of the object plane are also determined in this case. This is described in section 3.4.2.

Least-Squares solution If there are more than two co-planar objects, then the transformation is estimated using a least squares technique. Specifically, equation (3.9) is a linear constraint on the kernel vector (α, β, γ). If there are n objects

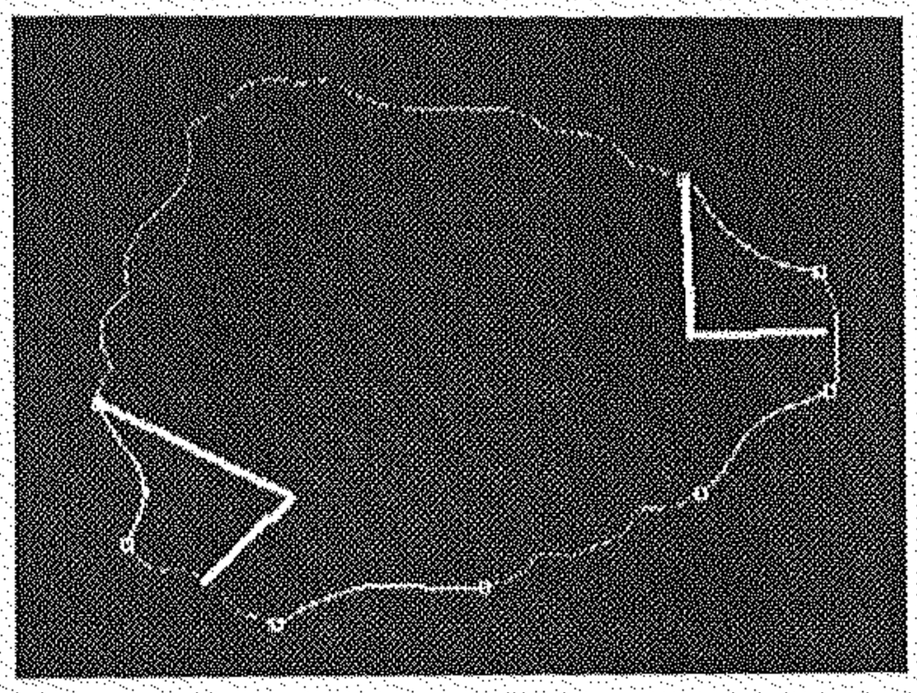

Figure 3.16: Local symmetries and matching concavities are extracted from figure 3.15. Note that local symmetries considered are neither parallel nor perpendicular.

Figure 3.17: Unskewed image of the object in figure 3.15.

Table 3.4: Angles between guiding vectors before and after unskewing the object in figure 3.15.

Objects	First symmetry	Second symmetry
Initial Angle	95.8	74.3
Final Angle	90.0	90.0

Table 3.5: Angles between guiding point triplet before and after unskewing for multiple objects in figure 3.18.

Objects	Spanner	Spoon	Hex spanner	Plier
Initial Angle	76.4	97.2	81.5	94.3
Final Angle	89.7	89.1	89.6	88.8

we seek the minimum of $||\mathbf{Mx}||^2$ subject to $||\mathbf{x}|| = 1$, where M is a $n \times 3$ matrix with each row given by equation (3.9). This is a standard problem in linear algebra. The solution is the eigenvector of $\mathbf{M}'\mathbf{M}$ with least eigenvalue. It is also possible to determine a covariance matrix for λ and μ in a similar manner to [7]. A more complete treatment of image noise and segmentation errors would weight each row of M according to a measure of its uncertainty.

Results of applying this least squares estimator to the affine scene in figure 3.18 are given in table 3.5 and figure 3.19. Note that the angle between the guiding vectors for the plier is not as good as the others due to the handle limitation discussed in section 3.3.1.

Back-projection is generally formulated as maximising a function - in this case one sensitive to the angle between back-projected guiding vectors, but unaffected by similarity transformations. Instead of the linear constraint given in equation (3.9) back-projection could be computed by minimising the nonlinear function $f(x) = \sum \cos^2 \theta_i$, (where θ_i is the skew angle between $\mathbf{a}^i$ and $\mathbf{b}^i$) and the sum includes all symmetric objects in the scene.

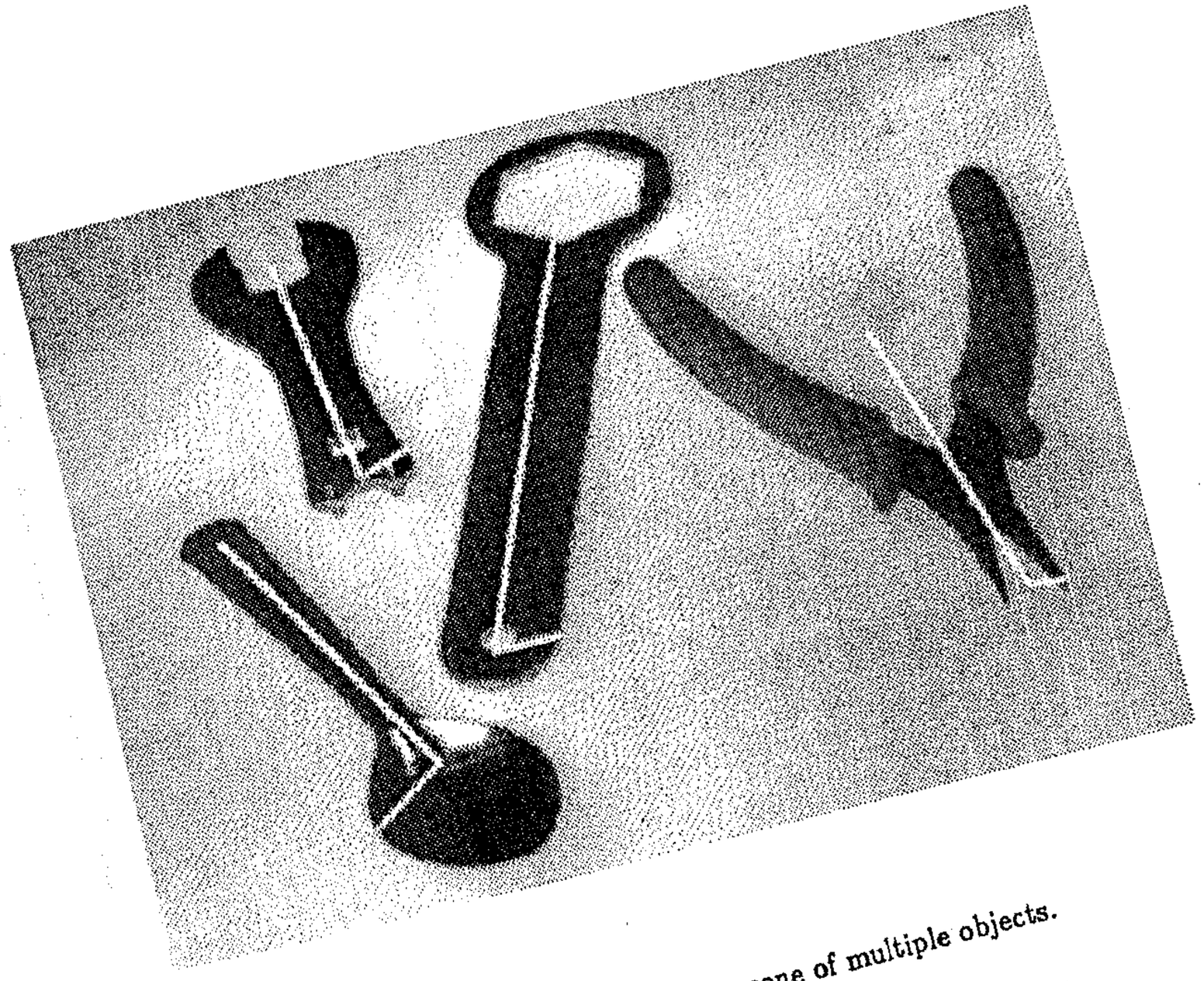

Figure 3.18: Affine scene of multiple objects.

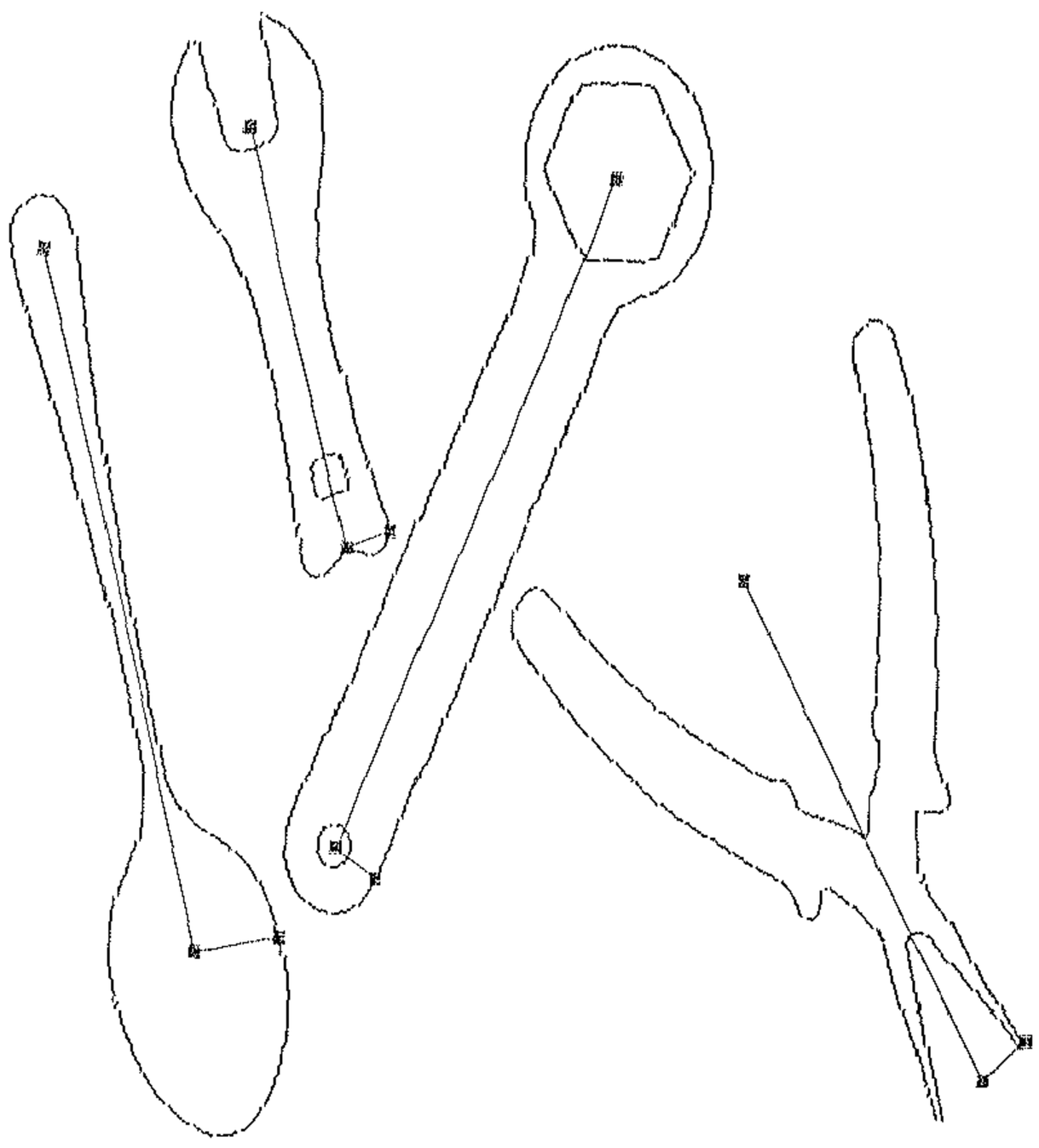

Figure 3.19: Unskewed image of multiple objects of figure 3.18.

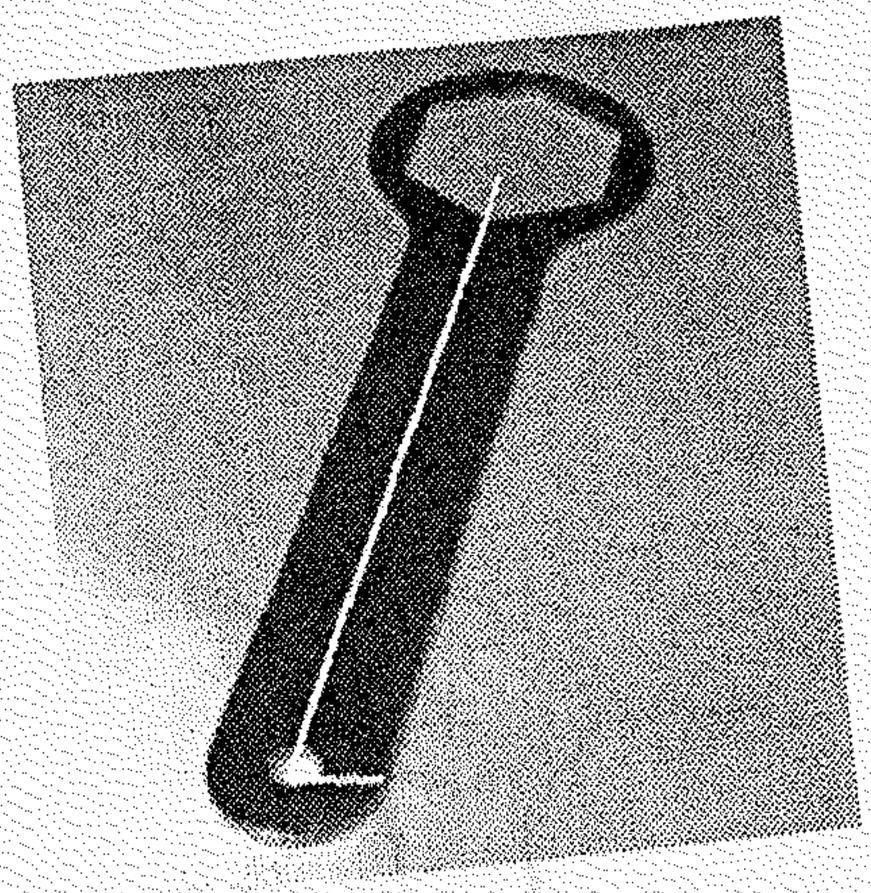

Figure 3.20: Skewed image of hex spanner with superimposed guiding vectors.

Combination with other constraints

If there is only one symmetry in the image this is only sufficient to determine the back-projection (modulo similarity) up to a one parameter family. However, other scene specific information can be incorporated to resolve the ambiguity.

In fact a number of constraints can be put forward in the case of hex spanner (figure 3.20), for example after unskewing all the sides of the hexagonal head should have equal length and the conic surrounding the hexagonal head should be circular.

More generally, for a regular isotropic shape, like a hexagon in this case, compactness ($(area)/(perimeter)^2$) as proposed by Brady and Yuille [13] and Horaud and Brady [36] or the weak isotropy condition as envisaged by [31] could be maximised in the object frame. The result of removing the ambiguity by imposing equality of the hexagonal sides is shown in figure 3.21.

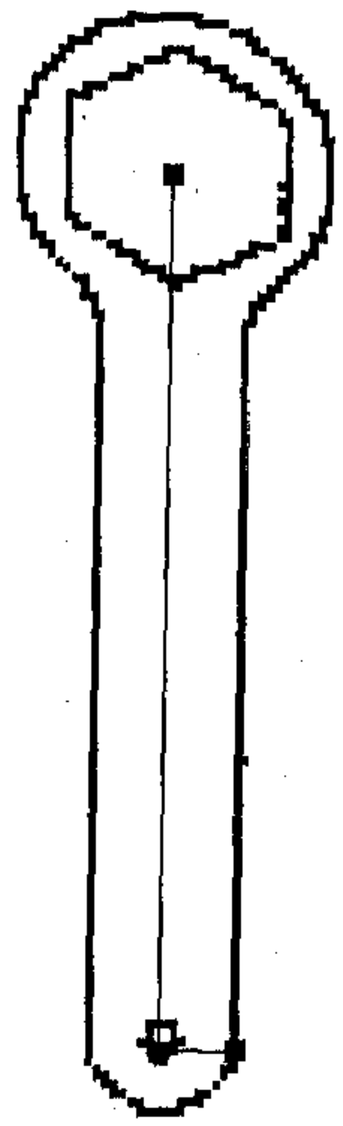

Figure 3.21: Unskewed image of hex spanner of figure 3.20.

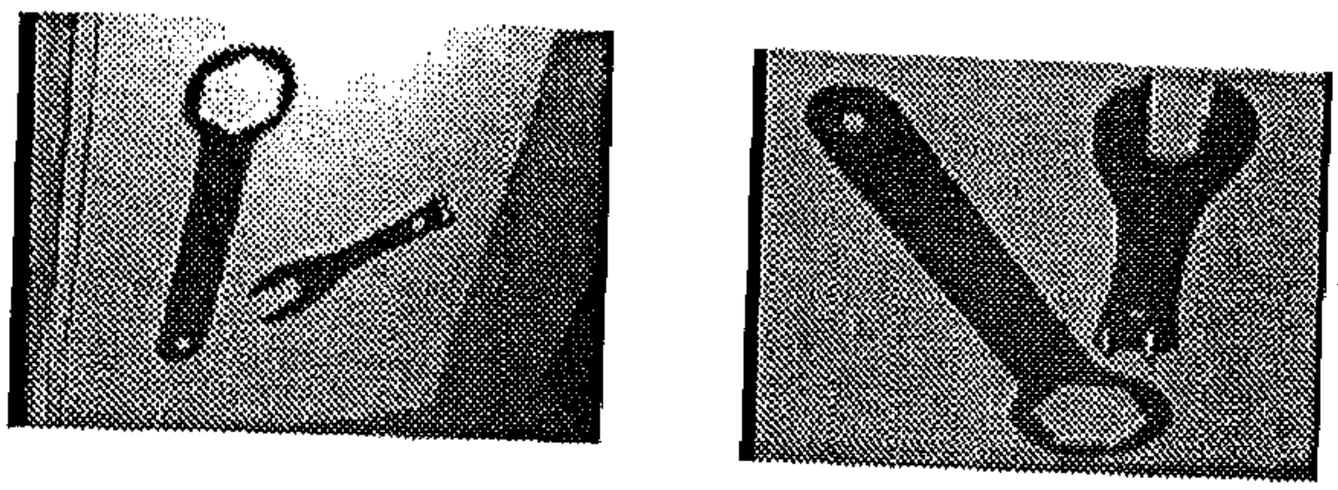

Figure 3.22: Examples of images used to calculate slant σ and tilt τ for each of the camera orientations. For each orientation σ and τ are recovered for three different object arrangements. Results are given in table 3.6.

3.4.2 Slant and tilt determination

If the camera aspect ratio is known, so that camera projection is scaled orthography rather than affine, then slant and tilt of the object plane can be determined from the ratio $\{\alpha : \beta : \gamma\}$. Refer to the lemma in section 3.2.2. Quite straightforwardly, the variables λ and r which appear in the back-projection operator (3.6) are respectively $\sec\sigma$, where σ is the slant of the object plane, and tilt (with $\lambda^2 = \lambda'^2$ so that $|\lambda| > 1$). See figure 2.3 of chapter 2 and [7]. The ratio $\{\alpha : \beta : \gamma\}$ determines λ up to sign, corresponding to a reflection of the plane. Thus slant is recovered up to the usual two fold ambiguity under scaled orthographic projection, i.e. σ and $\pi - \sigma$.

Calculated slant and tilts are given in table 3.6. The results are compared to (a) slant and tilt obtained by a method which back-projects a circle under perspective [63]; and (b) approximate measurements from the camera position. Two camera orientations are compared. Three results are given for each orientation corresponding to different arrangements of the co-planar objects. One representative image for each orientation is shown in figure 3.22.

3.4.3 Planarity tests

Suppose two symmetric planar objects are not co-planar; can this be detected from the image? If the objects are not co-planar then equations (3.12), or (3.18) may not have a solution. This provides a simple test for non-co-planarity which will always

Table 3.6: Slant and tilt values calculated from symmetry back- projection, circular back-projection and (approximate) camera position.

Orientation	Symmetry		Circle		Measured
	σ	τ	σ	τ	σ
figure 3.22 left	50 ± 3	88 ± 2	53	90	54
figure 3.22 right	41 ± 3	106 ± 4	47	118	44
figure 3.15	44	111	47	118	44

be passed if the objects *are* co-planar (subject to image noise), but non-co-planar objects may fail.

The test is derived from equation (3.12) as follows. We have

$$\frac{(1+\lambda^2)^2}{\lambda^2} = \frac{(\alpha+\gamma)^2}{\alpha\gamma-\beta^2} = 4\mu \tag{3.20}$$

Considering the λ equality first a number of constraints may be evaluated as follows:

1. For λ^2 to be real and positive, $\mu \geq 0$ (*from* $\mu \geq 0$, it follows that $\alpha\gamma > \beta^2$ or *det* $\mathtt{V} \geq 0$, where $\mathtt{V}$ is defined in equation (3.8)).
2. Multiplying out gives $\lambda^2 \pm 2\sqrt{\mu}\lambda + 1 = 0$. This only has real roots if the discriminant $(\mu - 1) \geq 0$, i.e. for λ real, $\mu \geq 1$.

This defines a region for acceptable $\{\alpha : \beta : \gamma\}$ solutions. If solutions do not lie in this region then the image cannot have arisen from co-planar objects.

An example is shown in figure 3.23, where the hex spanner is in a different plane from the other spanner. The calculated values of $\{\alpha : \beta : \gamma\}$ and μ are $\{-0.793 : -0.0920 : 1\}$ and -0.013. respectively. The above planarity test condition demonstrates that the objects are not in the same plane.

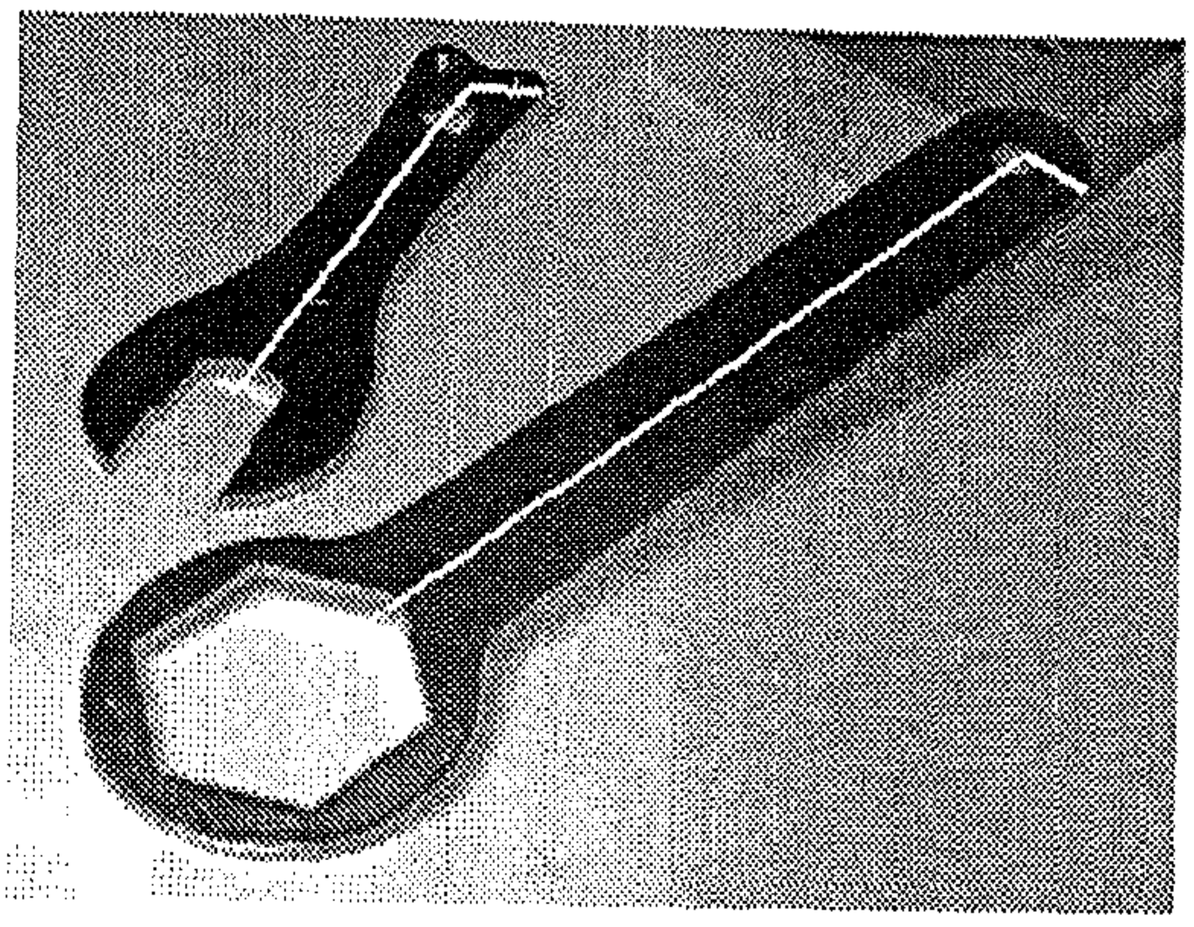

Figure 3.23: Affine image of two non-co-planar objects.

3.5 Discussions

We have demonstrated that the object relation of bilateral symmetry gives rise to image constraints that can be utilised in real applications. In particular bilateral symmetry restricts the affine transformation between corresponding image contours to a three dimensional subset of the planar affine group. This constraint allows these contours to be discriminated from other affinity related image curves.

As well as extending this idea to other object relations, like rotational symmetries, the approach can be extended to the most general planar object to image transformation - a projective transformation. Unlike the affine case, lines joining corresponding image points are not parallel in general, in projective images. How the symmetry restricts the transformation between corresponding image curves to a subset of the full projective group and how the projective invariants will be used to detect symmetry related contours are the key issues to be discussed in the next chapter.

Chapter 4

Symmetry in Projective Images

4.1 Introduction

In this chapter, we concentrate on the projective extensions of chapter 3. We assume the object to image transformation is approximated by planar projective group. Like affine case, since projective transformation form a group and a reflection due to a bilateral symmetry is a particular type of projective transformation, there may exist a *symmetry*, if one side of the contour could be mapped to the other *projectively* (8 dof). However, we show in section 4.2 that the image transformation is actually a subset of the plane projective transformation and has four degrees of freedom. These along with the use of projectively invariant properties are used in section 4.3 to develop an efficient algorithm for symmetry detection and verification.

The mathematical framework is established in section 4.2. The image to object back-projection is decomposed into two matrices - one responsible for the perspective and affine transformation (4 dof) and the other for similarity transform (4 dof). Since, we are interested in back-projection modulo similarity (translation, rotation and isotropic scaling), the necessary constraints are presented in section 4.2.2 to determine the perspective and affine distortion.

Under perspective as well as skewing there is perspective "fanning", see figure 4.1. The edge image of the top face of a cookie cutter in figure 4.1 is *unsuccessfully* unskewed using the back-projection algorithm detailed in chapter 3 [52], assuming

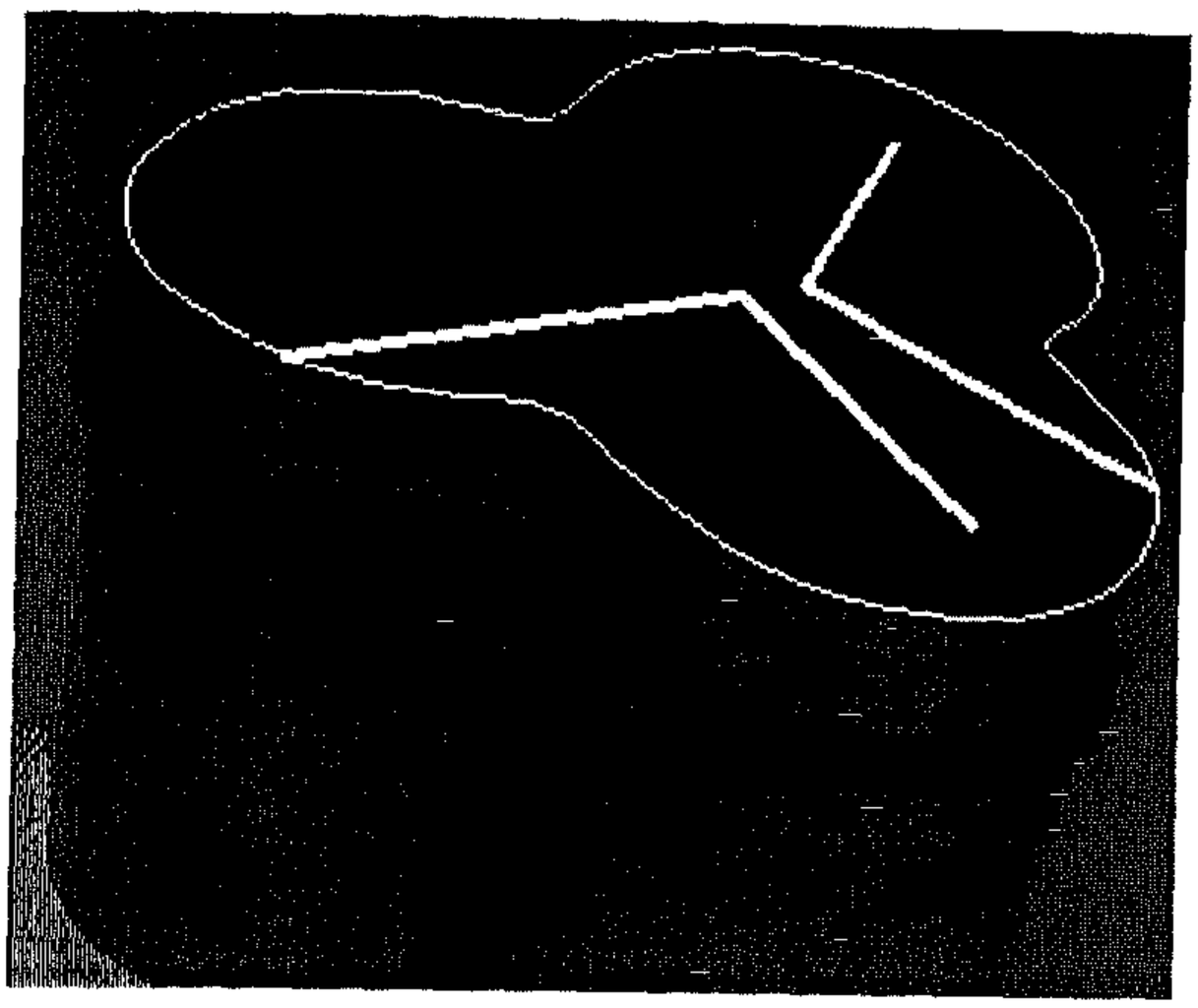

Figure 4.1: The projective image of a cookie cutter. The effect of perspective "fanning" is clear. The edge image of the top face of the object and a pair of guiding vectors are shown. Local symmetry line and the vector collinear to the direction joining corresponding points of a bilaterally symmetric contour are the guiding vectors.

image projection to be affine. This is shown in figure 4.2. The reader may care to pause at this point to view figure 4.18 which demonstrates an improved back-projection following methodologies developed in this chapter; however, the image projection, in this case, is assumed projective.

We have shown that two co-planar symmetries are needed, in general, to uniquely determine the "unskewing" matrix and the aspect ratio of the plane. Similar to affine case, as in chapter 3, if the camera aspect ratio is known, the back-projection matrix determines orientations of object plane relative to the camera. Finally, a test for non-co-planarity is given for two symmetric objects from a single image. The results are illustrated in section 4.4 with examples from images of real scenes followed by discussions.

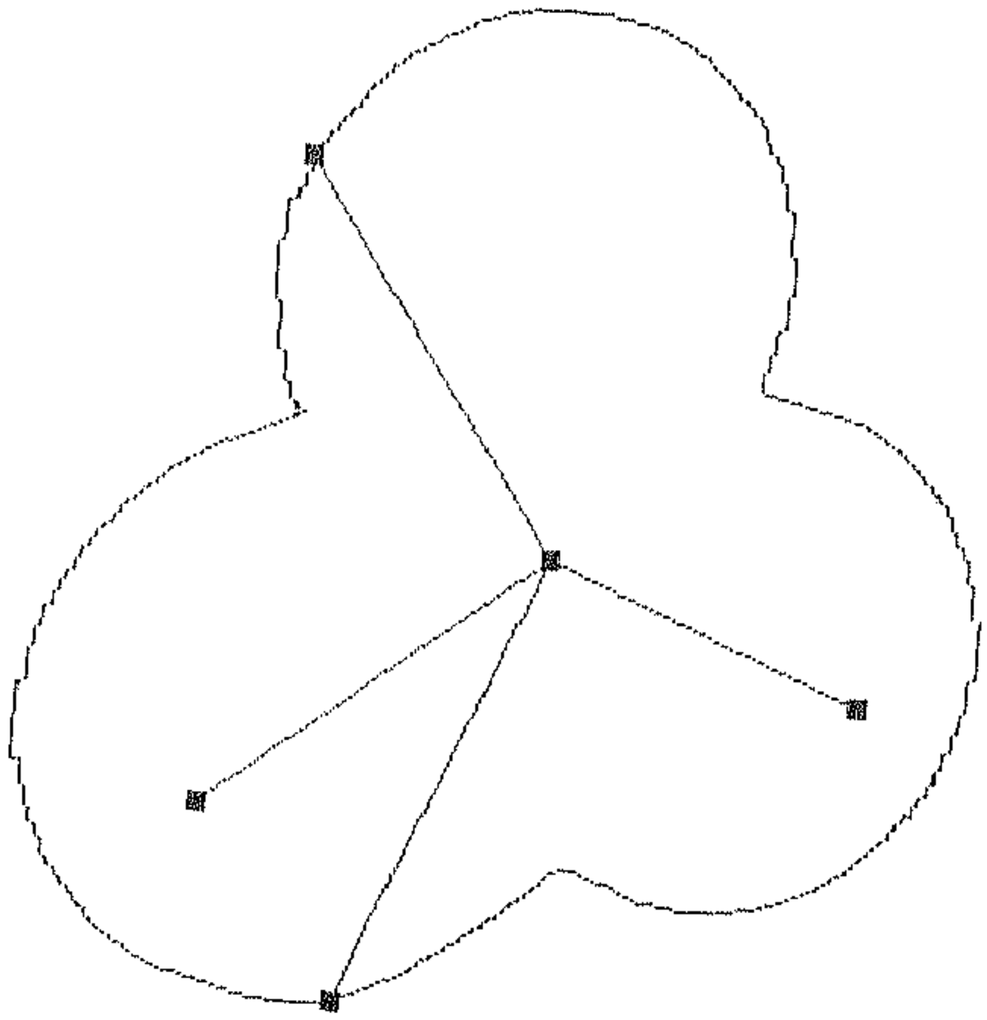

Figure 4.2: The edge image of figure 4.1 is unskewed following back-projection algorithm of chapter 3 [52] assuming image projection to be affine. Because of perspective "fanning", the image could not be unskewed exactly. The constraint enforced is that the guiding vectors are orthogonal in the object plane.

4.2 Mathematical framework

In the most general case there is a projective transformation between object and image planes. Figure 4.3 illustrates the situation under consideration: the image contours γ and γ' are the images, assumed to be under projective transformation, of two corresponding sides, Γ and Γ' respectively, of a planar object with a bilateral (mirror) symmetry. The key attributes of projectivity are properties like collinearity of points, intersection of lines and the cross ratio which are preserved. In particular, parallelism or length ratios on parallel lines as in the affine case are not preserved. For example in figure 4.3, mid-points of AA' and BB' give the symmetry line of the object; however, the projected symmetry line will not necessarily pass through the mid-points of aa' and bb'.

4.2.1 Image transformation

The following theorem explores the image transformation between the image curves γ and γ' in more detail. In particular, it fixes notation that will be used subsequently to back-project images.

Theorem 3 *Suppose two curves γ and γ', as in figure 4.3, are the images of two corresponding sides of a planar object with bilateral symmetry. Suppose further that image projection can be represented by a projective transformation. Then the transformation between γ and γ' has the following properties:*

1. *γ and γ' are related by a projective transformation. That is, if $\mathbf{x}$ is a point on γ then there is a point $\mathbf{x}'$ on γ' such that:*

$$\mathbf{x}' = \mathtt{T}\mathbf{x} \tag{4.1}$$

   where $\mathtt{T}$ is a non-singular 3×3 matrix, and $\mathbf{x}$ and $\mathbf{x}'$ are homogeneous three-vectors.

2. *The projective transformation $\mathtt{T}$ satisfies the following constraints:*

   (a) $\mathtt{T}^2 = k\mathtt{I}$, where k is a scalar.

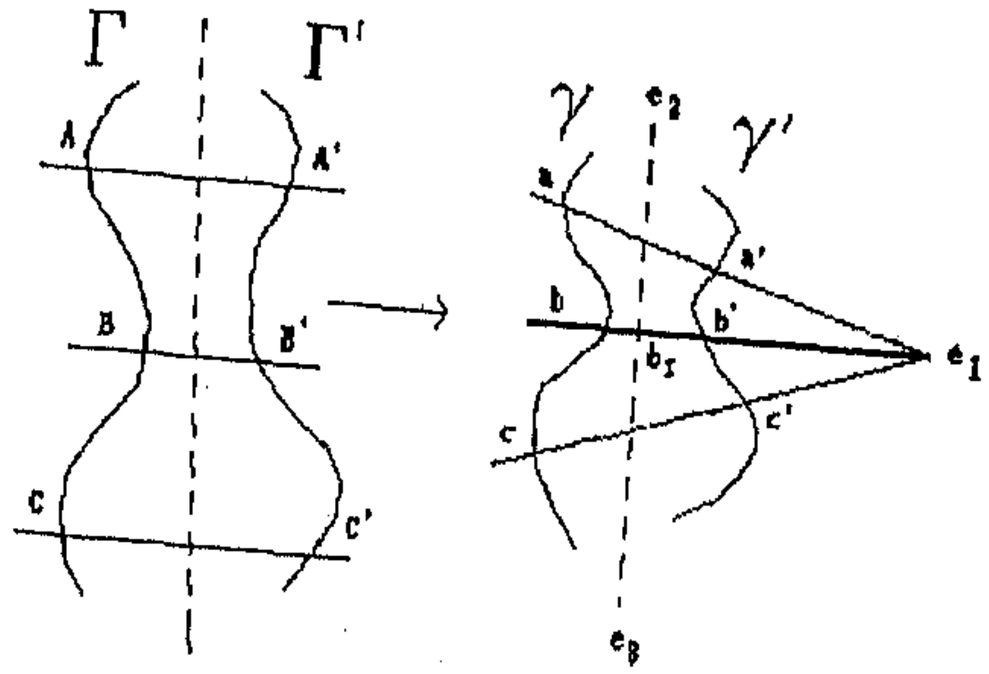

Figure 4.3: γ and γ' are the projective images of two corresponding sides of a planar object, Γ and Γ' respectively, with bilateral symmetry. Since projective transformation does not preserve parallelism and in particular mid-points, the imaged symmetry line may not pass through the mid-points of aa' and bb'. Following theorem 3, the symmetry axis is given by the line $(\mathbf{e}_2 \times \mathbf{e}_3)$. The eigenvector $\mathbf{e}_1$ is collinear to lines joining corresponding points of the imaged bilaterally symmetric object.

(b) The fixed points of T *are: a line of fixed points; and, a fixed point (not on the line) through which there is a pencil of fixed lines.*

A projection with these properties is a collineation of period two, also known as a 2 cyclic homography, and a planar harmonic homology [69].

3. *The matrix* T *has eigenvectors* $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. *Two of the eigenvalues, corresponding to* $\mathbf{e}_2$ *and* $\mathbf{e}_3$ *say, are equal. The third, corresponding to* $\mathbf{e}_1$ *is distinct and non-zero. The symmetry axis is given by the line* $(\mathbf{e}_2 \times \mathbf{e}_3)$. *Corresponding points,* $\mathbf{b}'$ *and* $\mathbf{b}$, *are collinear with* $\mathbf{e}_1$ *as shown in figure 4.3. The line* $\mathbf{b}'\mathbf{b}$ *intersects the symmetry axis in a point* $\mathbf{b}_I$ *say, and the four collinear points* $\mathbf{b}$, $\mathbf{b}_I$, $\mathbf{b}'$ *and* $\mathbf{e}_1$ *have a harmonic cross-ratio.*

4. *The transformation has four degrees of freedom. It can be determined from two correspondences.*

The proof is given in appendix A.2. Note that the projective transformation T is a non-singular 3×3 matrix with only four degrees of freedom instead of the usual

eight[1].

4.2.2 Back-projection

Following [52], in this section we consider the extent to which we can back-project images under projective transformation. The following analysis is similar to (affine) back-projection detailed in section 3.2.2; the extra difficulty, however, arises due to perspective "fanning". Note that step 3 of theorem 3 gives the direction $\mathbf{e}_1$ which is collinear to all the correspondence directions due to a bilateral symmetry. The vanishing point corresponds to $\mathbf{e}_1$. On the object corresponding points are collinear with this point at infinity since the lines joining corresponding points are parallel. Therefore, the notion of guiding vectors, as in chapter 3, could not be used in this case unless the perspective distortion is removed. We show in the following theorem that with two or more symmetries, the image could be uniquely back-projected *modulo similarity*.

Theorem 4 *Suppose we have an (un-calibrated) image of two co-planar symmetric objects then the image can be back-projected uniquely, modulo a similarity, provided that the two symmetry axes are neither parallel nor orthogonal in the object plane.*

Proof

First some notation for back-projections. Suppose the linear back-projection relating the object and image planes is given by:

$$\mathbf{X} = \mathtt{U}\mathbf{x} \tag{4.2}$$

where, $\mathbf{x}$ is the 2D image vector represented in the 3D homogeneous coordinate frame, $\mathbf{X}$ is the corresponding three-vector in the back-projected planar scene, and $\mathtt{U}$ is the 3×3 linear transformation matrix with $det\ \mathtt{U} \geq 0$ responsible for back-projection. Note that U defines a projective transformation with eight degrees of freedom.

Referring to theorem 3, the constraints for back-projection are:

[1] see chapter 2

1. *The intersection of the correspondence lines* ($\mathbf{e}_1$) *must be at infinity for parallelism in the back-projection. This is expressed projectively by requiring that* $\mathbf{e}_1$ *transform to a point* (q) *on the line at infinity,* $\mathbf{l}_\infty$.

2. *The correspondence direction* ($\mathbf{e}_1$) *and the symmetry line* ($\mathbf{e}_2 \times \mathbf{e}_3$) *must be perpendicular in the back-projection. This is expressed projectively by requiring that the intersection of the four points, namely, intersection of symmetry line and* $\mathbf{l}_\infty$, q, *and the two circular points* ($(1, i, 0)^t$ *and* $(1, -i, 0)^t$) *have a harmonic cross-ratio.*

The back-projection is achieved in two stages:

Stage 1. Removing the perspective distortion, leaving only affine.

Stage 2. Removing the affine distortion.

Each stage generates a 3×3 transformation matrix, and the back-projection matrix is the product of these two.

Stage 1. Removing the perspective distortion

Suppose there are two symmetries (a and b). Referring to theorem 3, the directions collinear to the corresponding points of the bilateral symmetries a and b, are given by $\mathbf{e}_1^a$ and $\mathbf{e}_1^b$ respectively. The line through these two points, say $\mathbf{l}'$, is given by,

$$\mathbf{l}' = \mathbf{e}_1^a \times \mathbf{e}_1^b = (l_1, l_2, l_3)^t \tag{4.3}$$

In removing the perspective distortion, we find the 3×3 transformation matrix, $\mathbf{T}_p$, which takes $\mathbf{l}'$ to $\mathbf{l}_\infty$. A point $\mathbf{p}_\infty$ on $\mathbf{l}_\infty$ must be of the form $\mathbf{p}_\infty = (p_\infty^x, p_\infty^y, 0)$. Note that if point transforms as $\mathbf{p}' = \mathbf{T}\mathbf{p}$, then line transforms as $\mathbf{l}' = \mathbf{T}^{-t}\mathbf{l}$. Hence, $\mathbf{T}_p$ is given by,

$$\mathbf{T}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(l_1/l_3) & -(l_2/l_3) & 1 \end{bmatrix} \tag{4.4}$$

Stage 2. Removing the affine distortion

The affine distortion is removed following back-projection algorithm presented in chapter 3 [52]. First, the transformation T_p is applied to $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ for symmetries **a** and **b**. Suppose, the transformed symmetry lines are given by $(\mathbf{e}_2^{a'} \times \mathbf{e}_3^{a'})$ and $(\mathbf{e}_2^{b'} \times \mathbf{e}_3^{b'})$ and the vectors representing symmetry correspondences are given by $\mathbf{e}_1^{a'}$ and $\mathbf{e}_1^{b'}$ for symmetries **a** and **b** respectively. By construction, $\mathbf{e}_1^{a'}$ (and, also $\mathbf{e}_1^{b'}$) is of the form $(p, q, 0)$ which is a line parallel to $y = (p/q)x$.

If, we express the symmetry line ($(\mathbf{e}_2^{a'} \times \mathbf{e}_3^{a'})$ or $(\mathbf{e}_2^{b'} \times \mathbf{e}_3^{b'})$) and the correspondence direction ($\mathbf{e}_1^{a'}$ or $\mathbf{e}_1^{b'}$) as 2D guiding vectors, say, **m** and **n** respectively, for any bilateral symmetry, these will back-project to vectors respectively parallel to and orthogonal to the symmetry axis. In the object plane (modulo similarity), therefore, the scalar product $(\mathbf{T}_a.\mathbf{m}).(\mathbf{T}_a.\mathbf{n}) = 0$, where, $\mathbf{T}_a$ is the 2×2 transformation matrix responsible for affine distortion. Consequently [52],

$$\mathbf{m}^t \mathbf{T}_a^t \mathbf{T}_a \mathbf{n} = 0. \tag{4.5}$$

The matrix $\mathbf{V} = \mathbf{T}_a^t \mathbf{T}_a$ is clearly symmetric and it is positive definite. Based on the methodologies developed in section 3.2.2, let the components of **V** be given by:

$$\mathbf{V} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}, \tag{4.6}$$

then we have from equation (4.5)

$$(m_x n_x \quad m_x n_y + m_y n_x \quad m_y n_y) \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = 0 \tag{4.7}$$

This is a linear constraint on α, β, γ. Two such constraints determine the *ratio* $\alpha : \beta : \gamma$. Referring to section 3.2.2, the sign is fixed by the requirement that **V** is positive definite, so that trace $\mathbf{V} = \alpha + \gamma \geq 0$.

Two symmetries generate two constraint equations (4.7):

$$\mathbf{M}\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix} = 0$$

where

$$\mathbf{M} = \begin{bmatrix} m_x^1 n_x^1 & m_x^1 n_y^1 + m_y^1 n_x^1 & m_y^1 n_y^1 \\ m_x^2 n_x^2 & m_x^2 n_y^2 + m_y^2 n_x^2 & m_y^2 n_y^2 \end{bmatrix} \tag{4.8}$$

Provided the matrix M is of rank 2 this uniquely determines the ratio $\{\alpha : \beta : \gamma\}$. It can be shown that M drops rank if any of the vectors $\{\mathbf{m}^1, \mathbf{n}^1, \mathbf{m}^2, \mathbf{n}^2\}$ are parallel, hence the clause in the theorem. Note that parallel lines in the object remain parallel in the image after affine transformation. □

The optimal solution when there are more than two symmetries or constraints is discussed in section 4.4. The back-projection matrices $\mathbf{T}_p$ and $\mathbf{T}_a$ could be obtained from linear least square solutions of equations (4.3) and (4.7).

Having determined $\mathbf{T}_p$ and $\mathbf{T}_a$, the composite back-projection matrix U is given by,

$$\mathbf{U} = \mathbf{T}_p.\mathbf{T}_a \tag{4.9}$$

where, $\mathbf{T}_a$ is a 3×3 matrix of the form

$$\mathbf{T}_a = \begin{bmatrix} u_{11} & u_{12} & 0 \\ u_{21} & u_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4.3 Detecting symmetries

We have followed the same methodologies as in chapter 3 to detect symmetry. Even though the projection ($\Gamma \rightarrow \gamma$ and $\Gamma' \rightarrow \gamma'$) is by a general projective transformation, as shown in section 4.2, the image curves γ and γ' are related by a (4 dof) planar projective subgroup. In this section, we describe how these results can be utilized to detect efficiently such image pairs and their reflectional symmetry axis.

The projective invariants are used as index functions which avoids the cost of an eight-dimensional (four-dimensional in our case) search over transformation param-

eters [64]. The three stages of an implemented algorithm are described in the following sections.

4.3.1 Generating and matching projective invariants

Two curves that are related by projective transformation have the same projective invariants. The converse is not necessarily true, but invariants can usefully be used to generate hypotheses for matching, which can subsequently be tested. Briefly, following the footsteps of affine invariants in section 3.3.1, a function $I(\Gamma)$ of a curve Γ is an *invariant* if $I(\gamma) = |\mathsf{U}|^w I(\Gamma)$ where γ is the image of Γ (refer to figure 4.3) under the projective transformation as defined in equation (4.2). The exponent w is the weight of the invariant. If $w = 0$ then the invariant is absolute, otherwise it is relative. Note that in order to determine local symmetries, the invariant cannot depend on global properties of the curve. Examples of (semi-local) projective invariants for smooth curves are given below.

Matching on invariants can be implemented as an $O(n)$ complexity process by the use of hashing (where n is the number of curves) [62,64]. We have implemented the simpler $O(n^2)$ algorithm, since n is small in the cases we have experimented with. It is straightforward to implement the more complicated algorithm.

Projective semi-local invariants

We use *canonical frame* to calculate projective invariants. Since, four points define a projective representation, we determine four stable *distinguished points* of each curve and map them to the four vertices of an unit square (Canonical frame with vertices at $(0,0)$, $(1,0)$, $(1,1)$ and $(0,1)$). We calculate area and moment of the transformed curve in the canonical frame. These are used as indexes to find the matching curve pairs. *Distinguished points* are curve "markers" that can be determined before and after a projective transformation. They include points preserved by projectivities (such as inflections, bi-tangent contact points, "cast" tangents), see figure 4.4. A number of examples on real images are shown in figures 4.6 and 4.9.

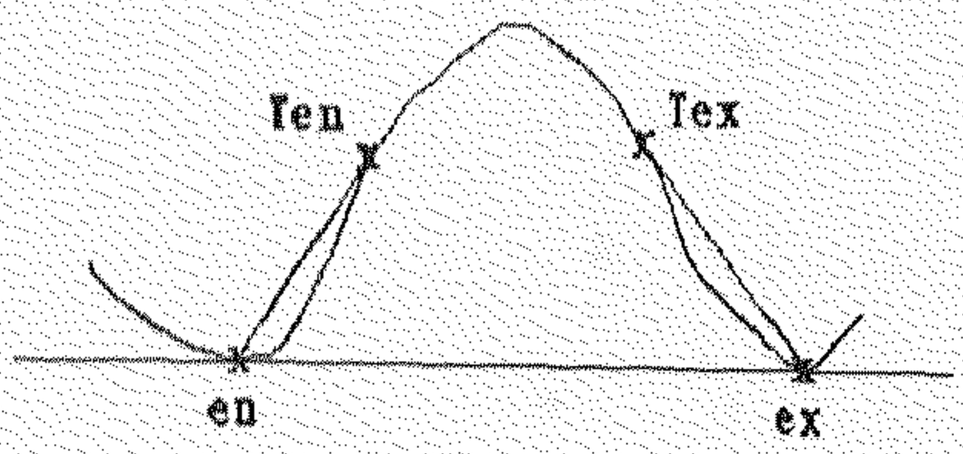

Figure 4.4: Examples of *distinguished points* for a non-convex curve. *en* and *ex* are concavity entrance and exit points and *Ten* and *Tex* are points on the curve which are tangent to a ray based at *en* and *ex* respectively.

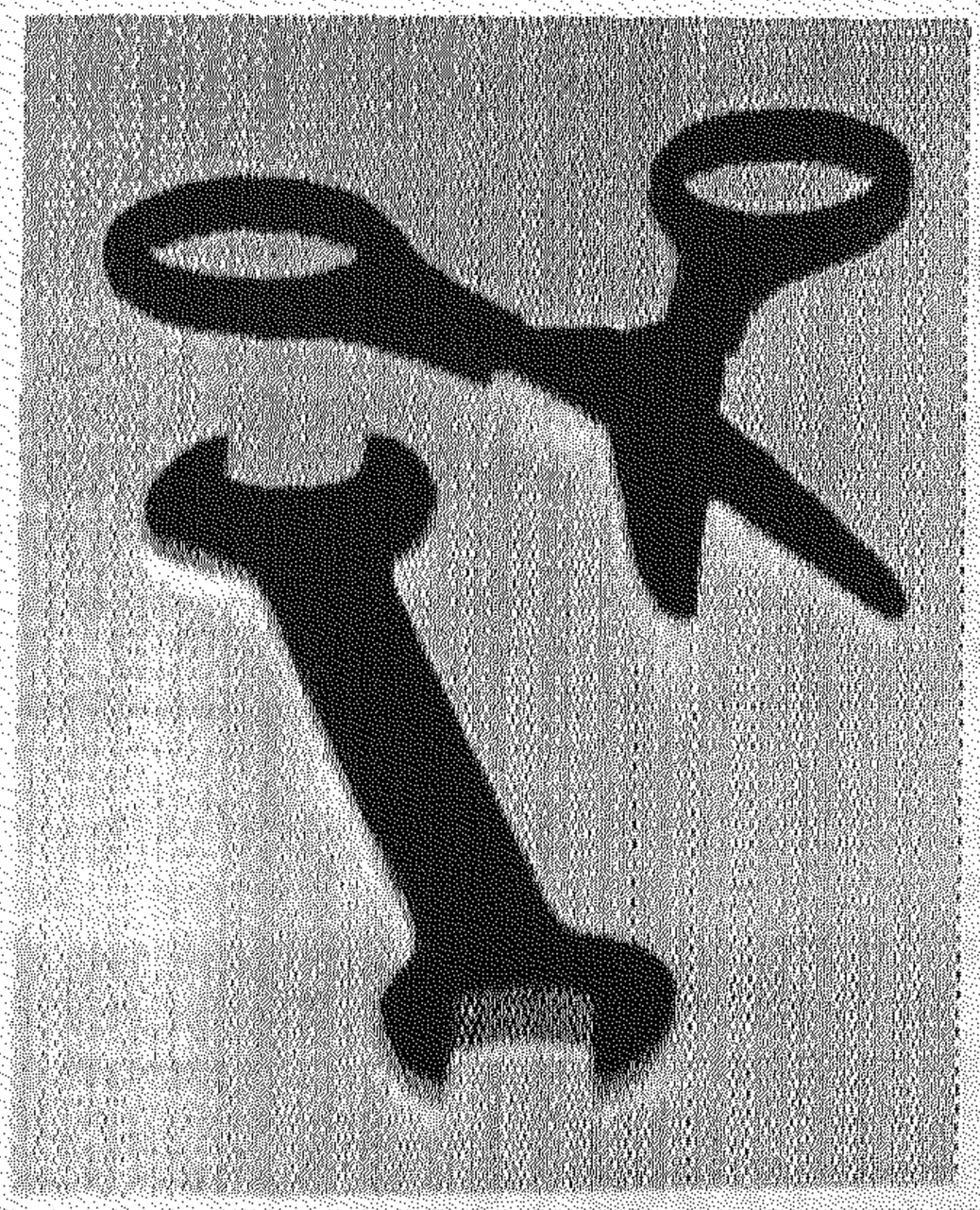

Figure 4.5: Projective image of a scene containing a spanner and a scissor.

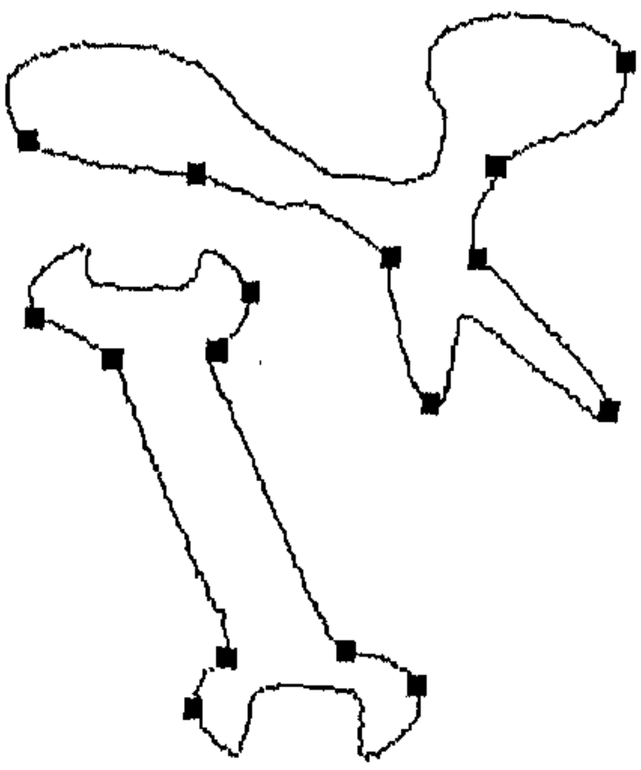

Figure 4.6: *Distinguished points* are marked on the projective scene of spanner and scissor of figure 4.5.

4.3.2 Determining the projective transformation

Having found two curves with matching invariant(s), the next stage is to determine if the curves are related by a projective transformation matrix T. This is achieved using *distinguished points* on each curve, and determining the transformation between these point sets. At least four matching points (two correspondences) are required to determine T. Note, points are ordered by the curve so the correspondence problem is greatly simplified.

If the curves are projectivity related then one side can be superimposed on the other applying transformation matrix T. An example is shown in figure 4.10. Note that accuracy of the reflection depends on number of extra correspondences (minimum two correspondences for 4 dof of T) between the matching curves. In that case, T is overdetermined and evaluated using pseudo-inverse technique which is detailed in section 4.3.4.

4.3.3 Verifying subset membership

As noted in section 4.2.1, if two curves are symmetry related then two correspondences (four matching points) are sufficient to determine the projective transformation (4 dof). If the transformation matrix does not obey the properties derived in section 4.2.1, theorem 3 in particular, then the two curves cannot be symmetry related and as a consequence inter-curve reflections, as in figure 4.10, will not work. When more points are available, the form of the transformation is used as a constraint (via a Lagrange multiplier) in a least squares estimator. Details are given in the next section.

4.3.4 Implementation and results

Feature extraction Image contours are extracted using a local implementation of the Canny [15] edge detector. For a non-convex curve, following Rothwell *et al* [63], we exploit concavities by constructing a bi-tangent across the concavity calculating convex hull to the edge point set. Note that the point of tangency and bi-tangent line are covariant to projection. If the length of the bi-tangent line and its maximum distance from the curve it is encompassing (concavity height) are greater than some thresholds, then the two points of tangency constitute entrance and exit points of a *significant concavity*.

The next step is to find further two points of tangency to the curve using tangent lines starting from entrance points of the concavity (cast tangents). These points can also be evaluated as before by constructing convex hull to the *significant concavity* only. These view-independent four stable points constitute a distinguished frame which can be mapped to a canonical frame. The four distinguished points or projective basis points of the *significant concavities* of the spanner and the scissor (figure 4.5) are shown in figure 4.6. Figure 4.7 shows the *significant concavities* of the spanner of figure 4.5 when mapped to the canonical frame.

For a convex curve segment, semi-differential invariants [4] may be used by fitting b-spline to the curve segment and determining curve inflection points.

Concavity matching For each closed contour in the scene, matched concavities are detected using projective invariant indexes (area and moment of concavities

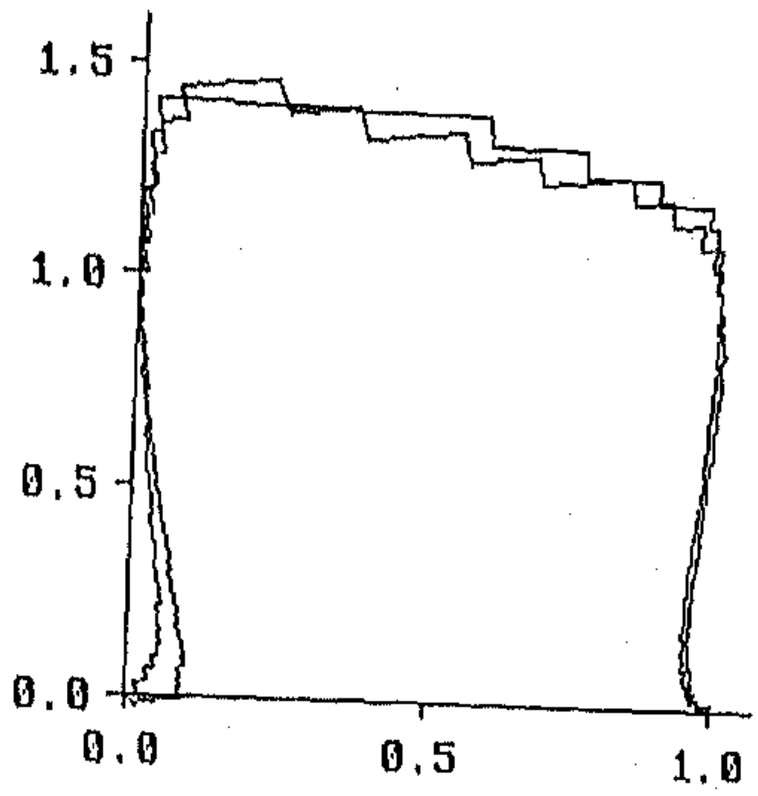

Figure 4.7: The transformed curves of the two *significant concavities* of the spanner in figure 4.5 are shown when mapped to the canonical frame with co-ordinates at $(0,0)$, $(1,0)$, $(1,1)$ and $(0,1)$.

Table 4.1: Projective invariant indexes of significant concavities of figure 4.9. M_x and M_y are the moments about x and y axes respectively.

Invariant	*Cycle spanner*		Scissor		Hex spanner		Spanner	
Area (Canonical frame)	1.34	1.41	1.49	1.61	2.10	2.12	1.13	1.18
M_x (Canonical frame)	.572	.564	.396	.471	.630	.594	.327	.38
M_y (Canonical frame)	1.712	1.663	1.931	1.81	2.326	2.277	1.469	1.396

mapped to the canonical frame) as described in section 4.3.1. Corresponding points in the matched concavity pair are determined from the tracing order (clockwise or anti-clockwise) of the image contour from which concavities are extracted. These point correspondences are used to determine the projective transformation between corresponding curves.

Figure 4.9 shows the *significant matching concavities* extracted from the locally symmetric objects of figure 4.8. The concavity distinguished points are marked in figure 4.9. Projective invariant indexes used to match the concavities are given in the table 4.1.

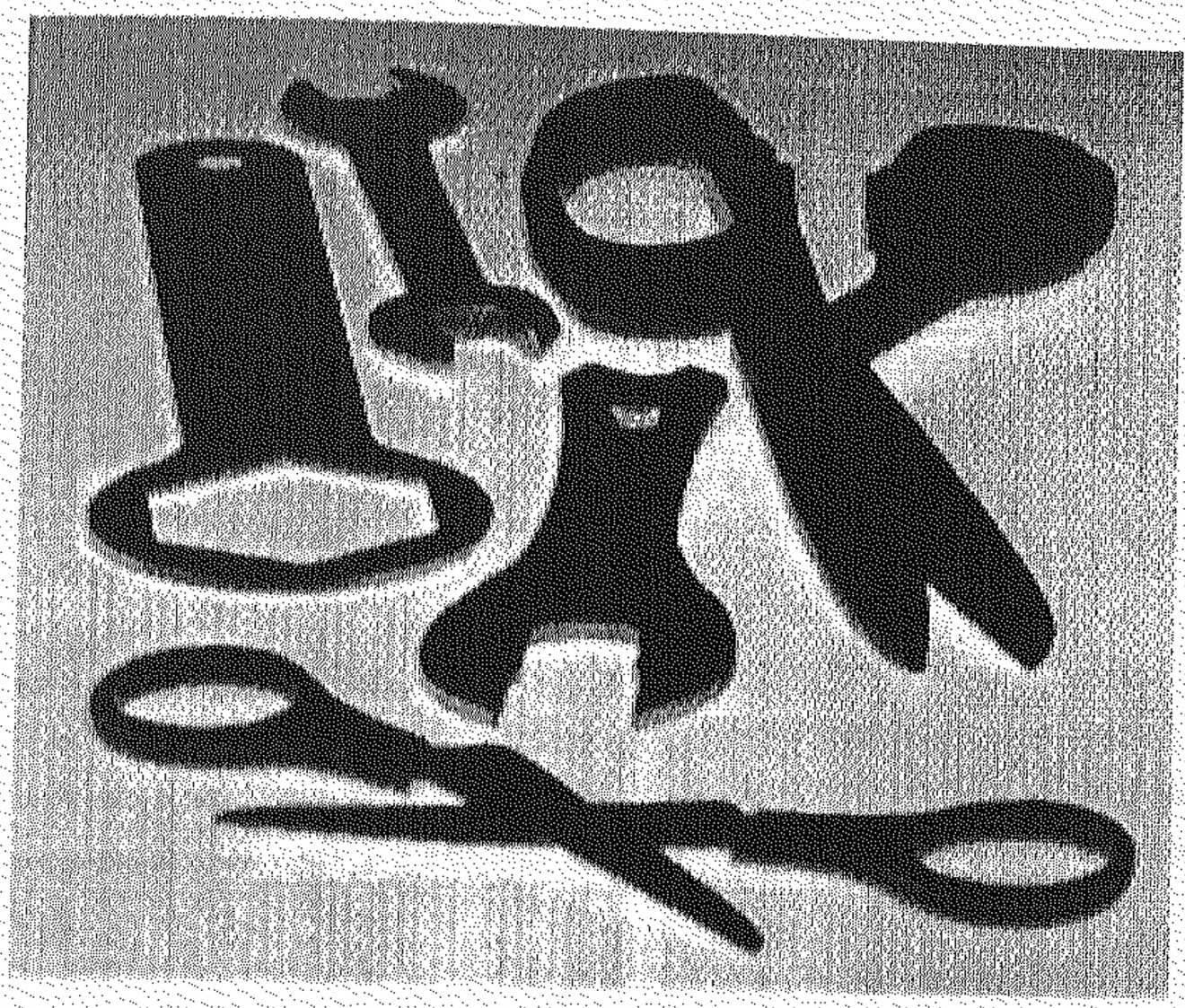

Figure 4.8: Projective scene of symmetric and non-symmetric objects.

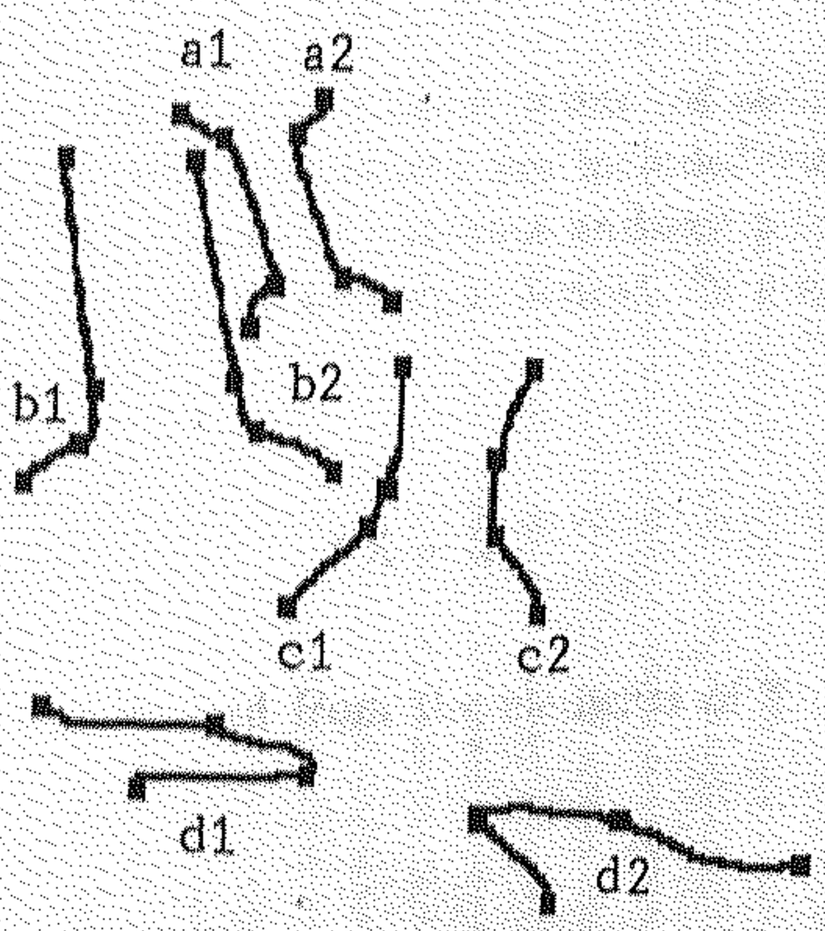

Figure 4.9: Matched concavities $(a1, a2)$, $(b1, b2)$, $(c1, c2)$ and $(d1, d2)$ are extracted from the symmetric objects of figure 4.8. The *distinguished points* are marked. Clearly $d2$ is not an imaged symmetry of $d1$.

Table 4.2: Improvement of projective transform T using 4-point correspondence for the spanner in figure 4.10. Note that, referring to theorem 3, two of the eigenvalues of T (corresponding to e_2 and e_3) should be identical.

Using 2-point correspondence		Using 4-point correspondence	
eigenvalue	corresponding eigenvector	eigenvalue	corresponding eigenvector
-0.9793	{-1.6681, 0.5431, 0.0}	-1.0000	{-1.6792, 0.5431, 0.0}
1.018	{-0.6982, -0.9172, 0.0}	1.0001	{-0.7311, -0.9544, 0.0}
0.9884	{12.9872, -3.8431, 1.0}	1.0000	{13.1844, -3.9740, 1.0}

Projective transform The next step is to determine the projective transformation T arises from an imaged reflectional symmetry between the matching concavities. Necessary equations are detailed in section 4.2.1. Since, there is a bilateral (reflective) symmetry, it can be shown that a necessary and sufficient test is that

$$T^2 = \lambda I \tag{4.10}$$

In this way the non-symmetric object in figure 4.8 is eliminated as shown in figure 4.9.

In practice, T determined from the two basis points (of each concavity) is not sufficiently accurate; so extra correspondences are included via pseudo-inverse. The bi-tangent points (other than concavity entrance and exit points) determined in feature extraction stage provide extra correspondences. In straightforward notation, equation (4.1) is rewritten as

$$PX = Q \tag{4.11}$$

where X is an eight-vector formed from the elements of T. This is solved using a pseudo-inverse as

$$X = P^t(PP^t)^{-1}Q \tag{4.12}$$

Figure 4.10 proves the point where one side of the symmetry axis of the object is reflected to the other using T evaluated applying pseudo-inverse. Table 4.2 shows the improvement in projective transform T using pseudo- inverse.

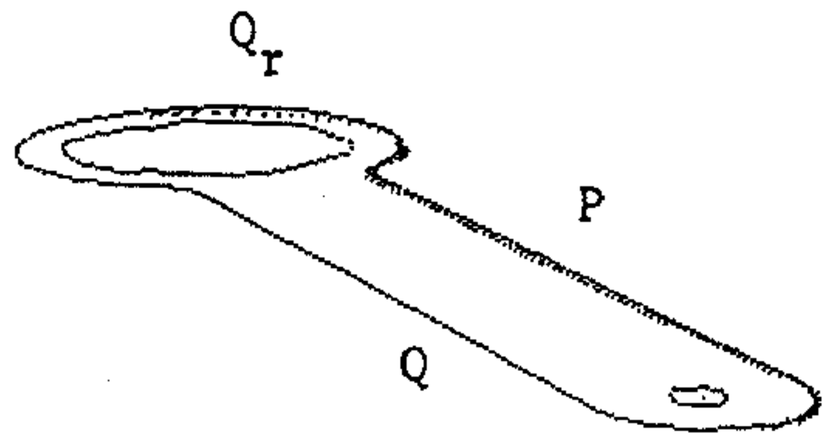

Figure 4.10: The curve Q_r is the "reflection" of the curve Q using projective transformation computed from four point correspondences.

An alternative method for improving the accuracy of the projective transformation is to minimise differences between the curve on one side and the other side superimposed or reflected (so it should be identical). This is done with the initial estimation of matrix T and improving it (T) iteratively. The steps involved are as follows:

1. Evaluate an approximate T from two point correspondences.

2. For a bilateral symmetry, say γ and γ' as in figure 4.3, one side is reflected to the other (as Q_r, reflection of Q, is mapped to the other side P in figure 4.10) using T evaluated in step 1.

3. For a number of points along the curve P, say every 5th edgel, measure the perpendicular distance between P and Q_r. Vertical Scanning from the selected points of P gives the corresponding points of Q_r.

4. Iterate to minimise the sum of squared perpendicular distance using Levenberg-Marquardt method [57]. For N points along P, a χ^2 merit function is built

$$\chi^2(\mathbf{a}) = \sum_{i=1}^{N} [\mathbf{x}_P - \mathrm{T}\mathbf{x}_Q]^2$$

where **a** is an eight-vector formed from the elements of (8 dof) T. $\mathbf{x}_P$ and $\mathbf{x}_Q$ are points along P and Q respectively. The merit function is minimised from the gradients of χ^2 with respect to parameters **a**. The detail methodology is provided in [57]. An example is shown in figure 4.11.

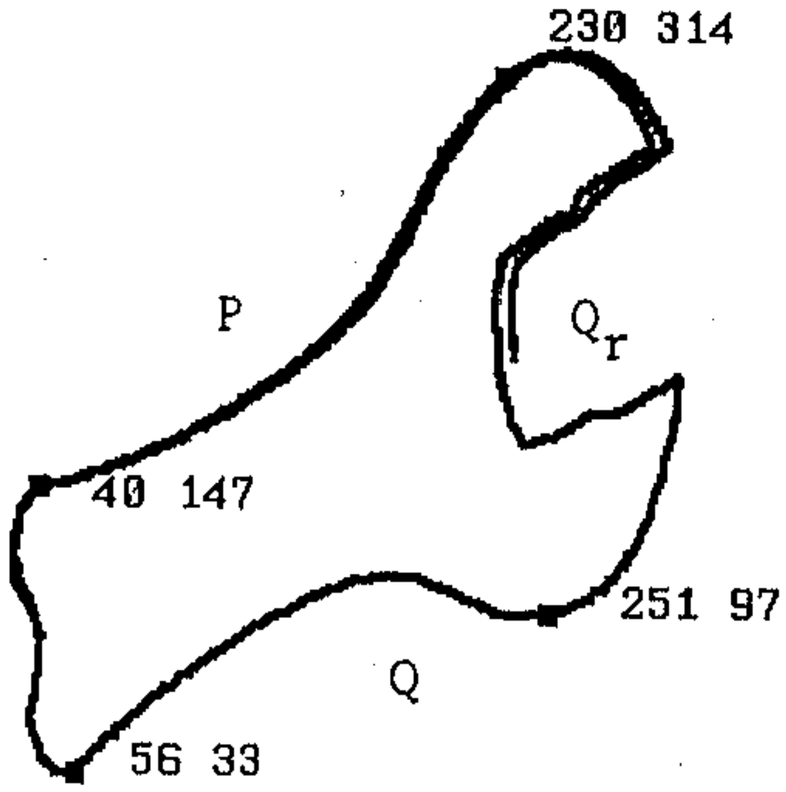

Figure 4.11: Example of use of improved T using iterative method. The *Qr* is the reflection of *Q* on *P*. The co-ordinates of two point correspondences, used for initial estimation of T, are shown.

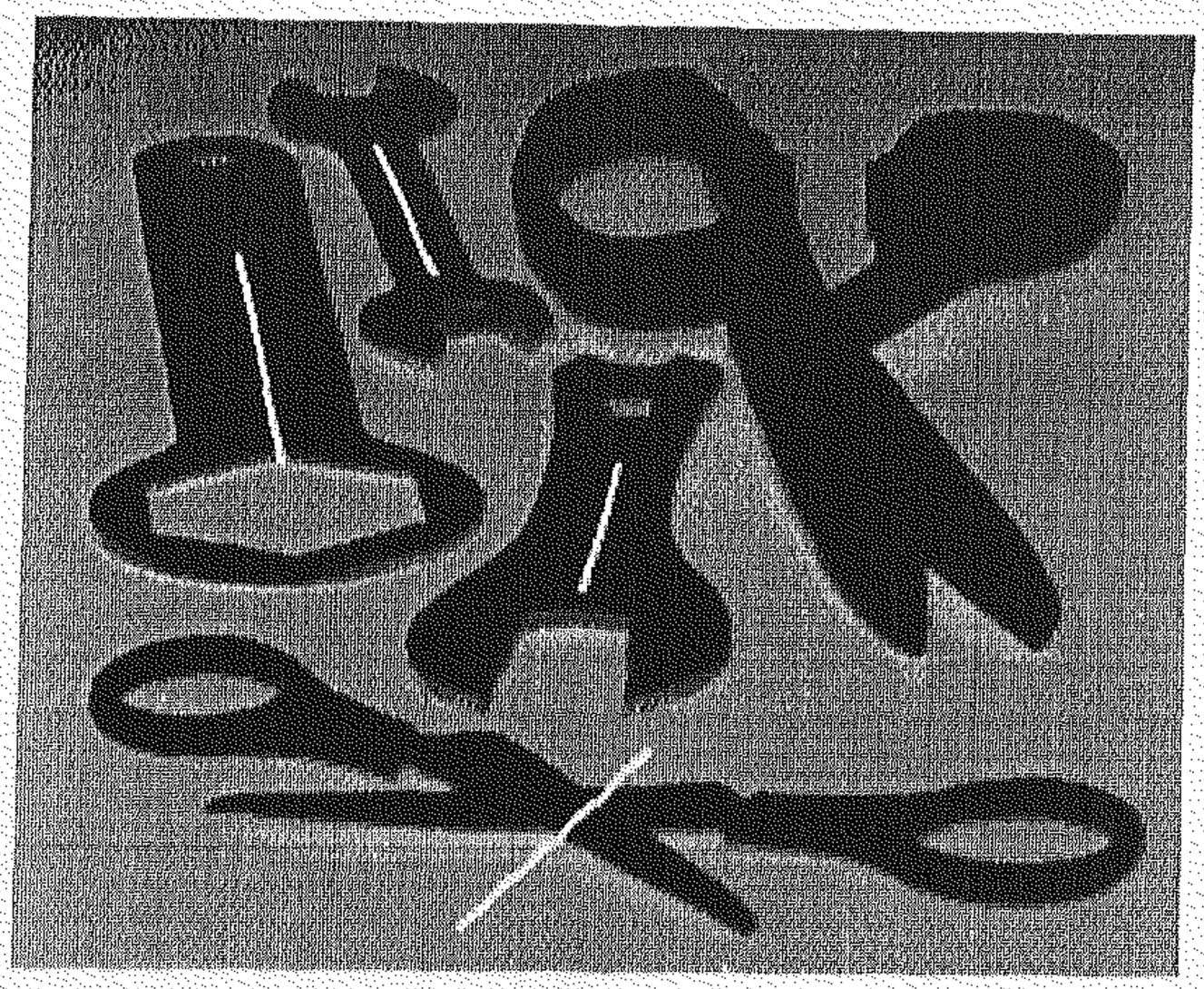

Figure 4.12: Local symmetry lines are drawn only on the symmetric objects. This eliminates the projectively related contours of the upper-right object in figure 4.8.

Projective symmetry axis Having determined the projective transformation, the symmetry axis is drawn as described in section 4.2.1. The step 3 of theorem 3 provides an "optimal" algorithm for extracting the projective symmetry axis - namely determine a T which best maps one side of the contour to the other, and compute the axis from the eigenvectors. In figure 4.12, local symmetry lines are drawn correctly on the objects of figure 4.8 using above algorithm.

An alternative simple method arises by noting that any projectively covariant construction[2] on the object which generates points on the symmetry axis, can be used to determine points on the symmetry axis in the image. For example, find corresponding pairs of distinguished points on each side of the outline, say a corresponding to a', b corresponding to b'. Then line pairs $\{ab, a'b'\}$ and $\{ab', a'b\}$ both intersect on the symmetry axis. These two intersections, say s_1 and s_2, determine the projective symmetry axis. In the absence of measurement noise both the "optimal" and simple method will produce the same line. Figure 4.13 demonstrates the

[2]A projectively covariant construction is where the same result is obtained if the construction is performed in one frame, and then the result projected to a new frame, or if the construction is performed in the new frame on the projection of the original curves; constructions with this property are based around incidence and counting properties. For example, a tangent line is a covariant construction.

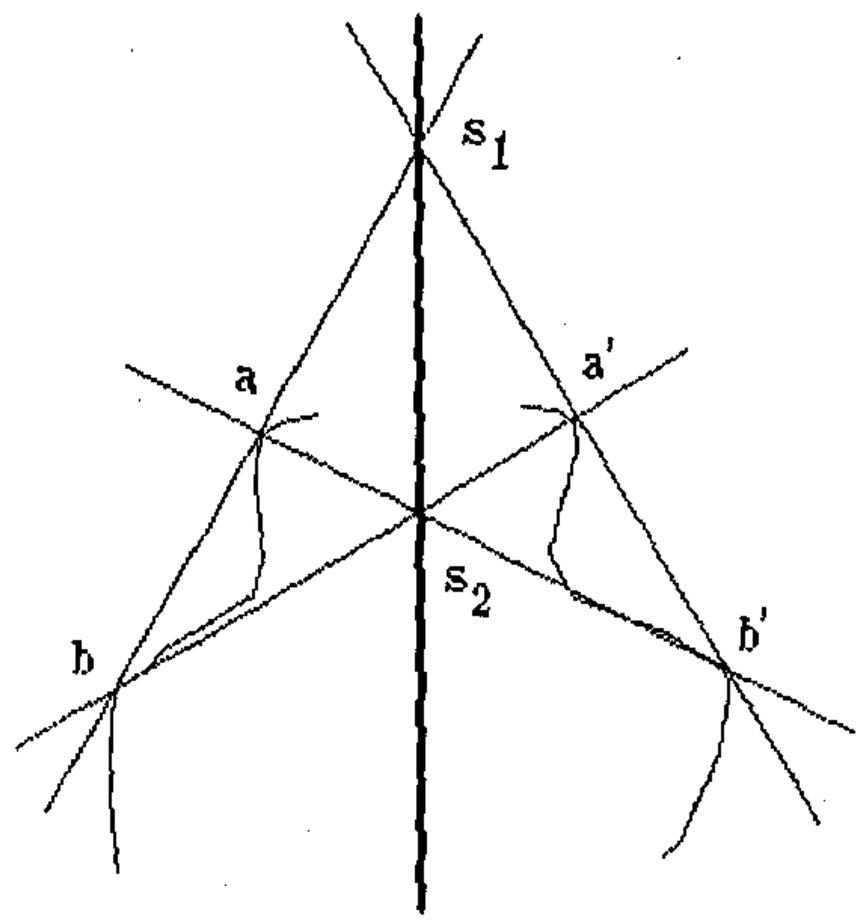

Figure 4.13: Symmetry detection by construction. As the intersection points are preserved *under projectivity*, the line joining s_1 and s_2 gives the symmetry line.

construction.

From the properties of the projective transformation matrix T, representing reflection, the symmetry axis is also given by the eigenvectors of $\mathtt{T}^{-t}$. To see this, remember that if point transforms as $\mathbf{x} = \mathtt{T}\mathbf{X}$, then a line transforms as $\mathbf{l} = \mathtt{T}^{-t}\mathbf{L} = \lambda\mathbf{l}$. Note that two (non-zero) eigenvectors represent the symmetry axis and the line joining corresponding pairs of points, *viz.* aa' and bb' (as in figure 4.13) respectively.

Table 4.3: Angles between the symmetry axes and correspondence directions before and after unskewing for objects in figure 4.14.

Objects	Spanner	Opener
Initial Angle	93.4	91.7
Final Angle	90.0	90.0

4.4 Applications

4.4.1 Back-projection

Here we determine the back-projection of the image to the object plane using the results of section 4.2. Note back-projection does not require camera aspect ratio (or any of the camera intrinsic parameters).

Accurate back-projection requires accurate determination of the eigenvectors of the matrix T (refer theorem 3). In practice we find that four point correspondences and use of a pseudo-inverse, following equation (4.12) are sufficient to determine T to a satisfactory accuracy.

After removing perspective distortion (refer to stage 1 of section 4.2.2), the guiding vectors can be mapped to the unskewing frame, an intuitive and simple construction detailed in chapter 3. Also see figure 3.12. The unskewing of affine distortion (refer to stage 2 of section 4.2.2) could be achieved following back-projection algorithm of section 3.4.1. As mentioned in chapter 3, if there are more than two symmetries, where a least squared solution is required, then the above method is not easily generalisable. However, the formulation of section 4.2.2 is not restricted. Its application in a least-squared solution is described below.

Figure 4.14 is a scene consisting of two locally symmetric objects. The result of unskewing the objects of figure 4.14 is shown in figure 4.15. Table 4.3 gives the angles between the symmetry axes and correspondence directions before and after unskewing the objects of figure 4.14.

Similar to affine case in section 3.4.1, if an object contains several local symmetries, the object alone is sufficient to determine the back-projection (provided the usual conditions are satisfied).

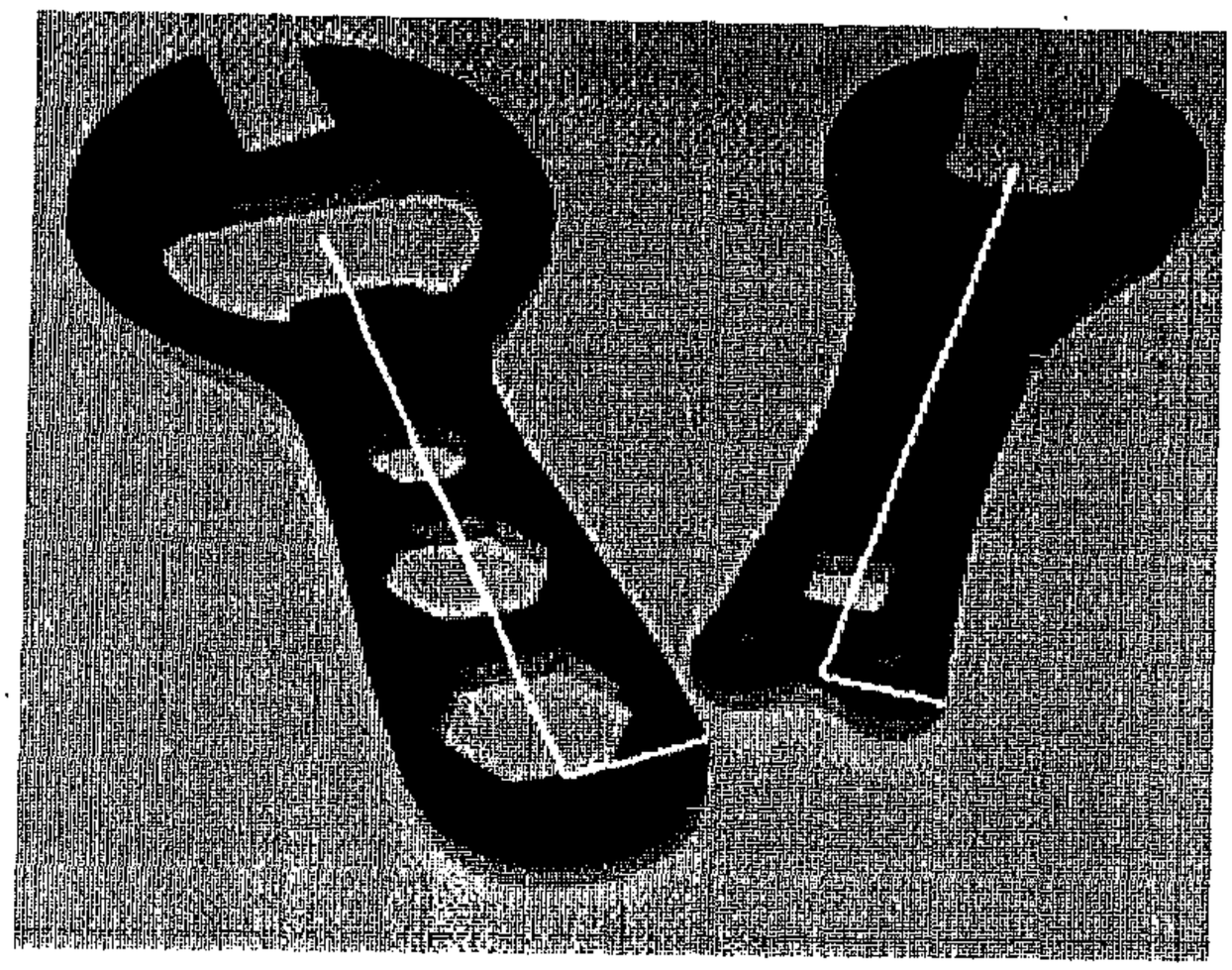

Figure 4.14: Projective view of two objects with superimposed symmetry axes and correspondence directions.

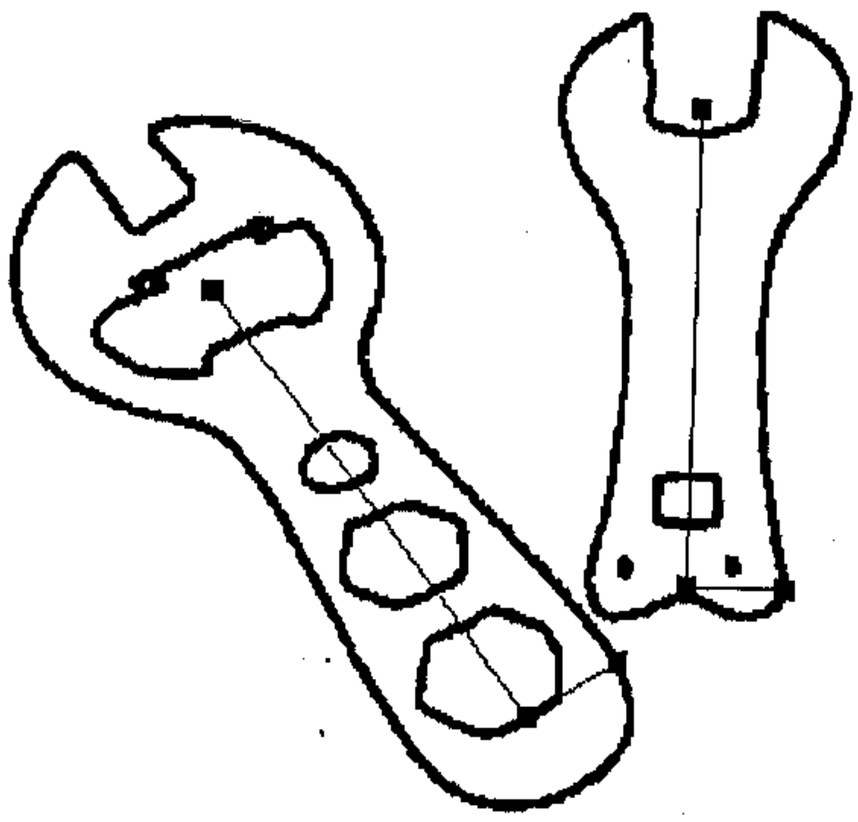

Figure 4.15: Back-projected image of the objects of figure 4.14.

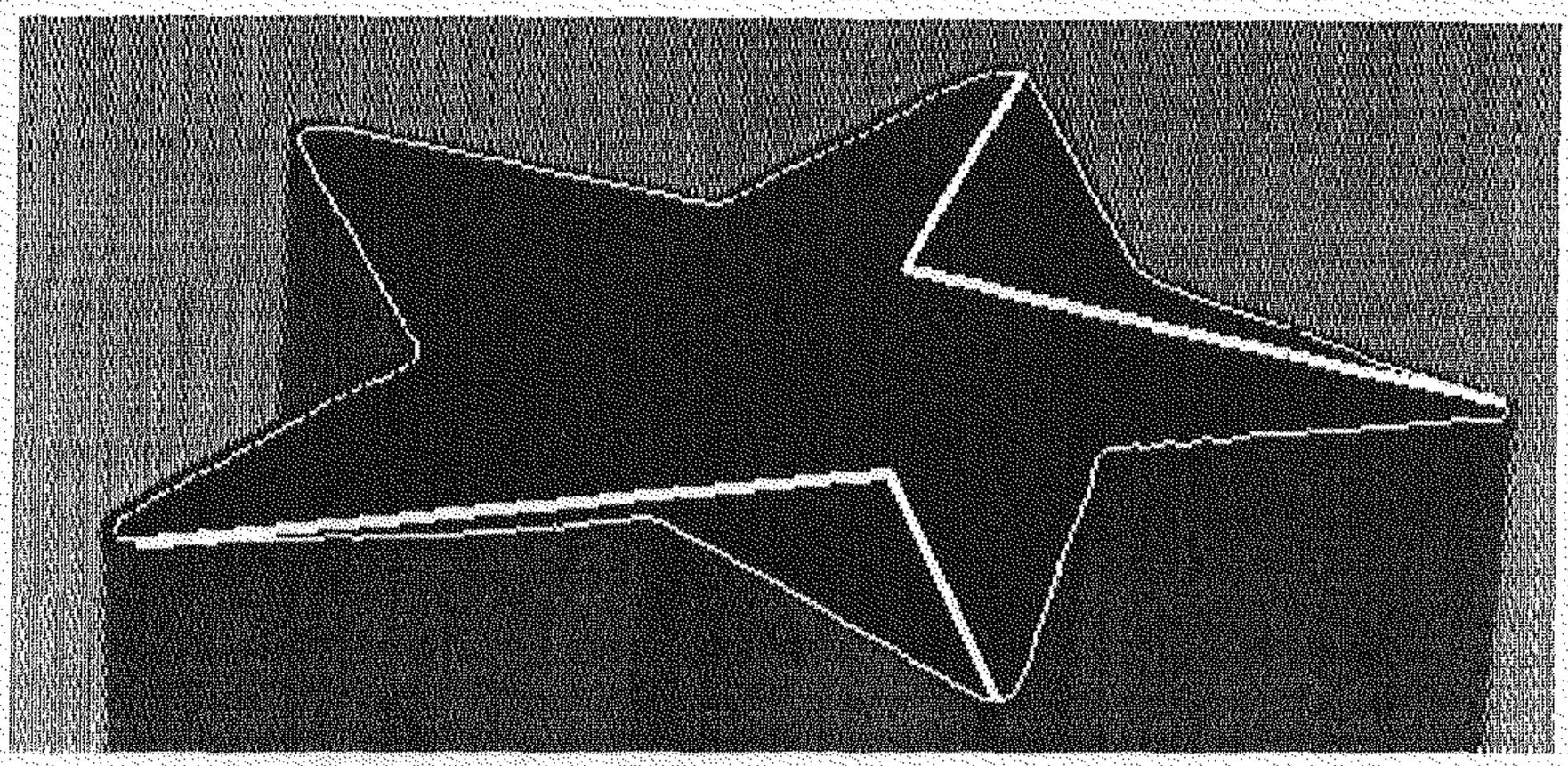

Figure 4.16: Projective image of an object with multiple local symmetries. The edge of the star-faced cookie cutter and its symmetry axes are shown. Note that local symmetries considered are neither parallel nor perpendicular in the object plane.

Table 4.4: Angles between the symmetry axes and correspondence directions before and after unskewing the object of figure 4.16.

Objects	First symmetry	Second symmetry
Initial Angle	87.3	91.02
Final Angle	90.0	89.97

The edge image and the the symmetry axes and correspondence directions of an object with multiple symmetry is shown in figure 4.16. We have taken two local symmetries of the top face of the star-shaped cookie cutter. Figure 4.17 shows the back-projected edge image and table 4.4 demonstrates the back-projection. Also, see figure 4.18 which demonstrates an accurate back-projection of the edge image of the object of figure 4.1. This is achieved after removing both perspective and affine distortion compared to figure 4.2, where only affine distortion is removed.

Least-squares solution If there are several local symmetries present, as in a real scene shown in figure 4.19, then the transformation is estimated using a least squares technique. Similar approach is detailed in section 3.4.1. To find, T_p as in equation (4.4), we have to find the line l' which best satisfy all the correspon-

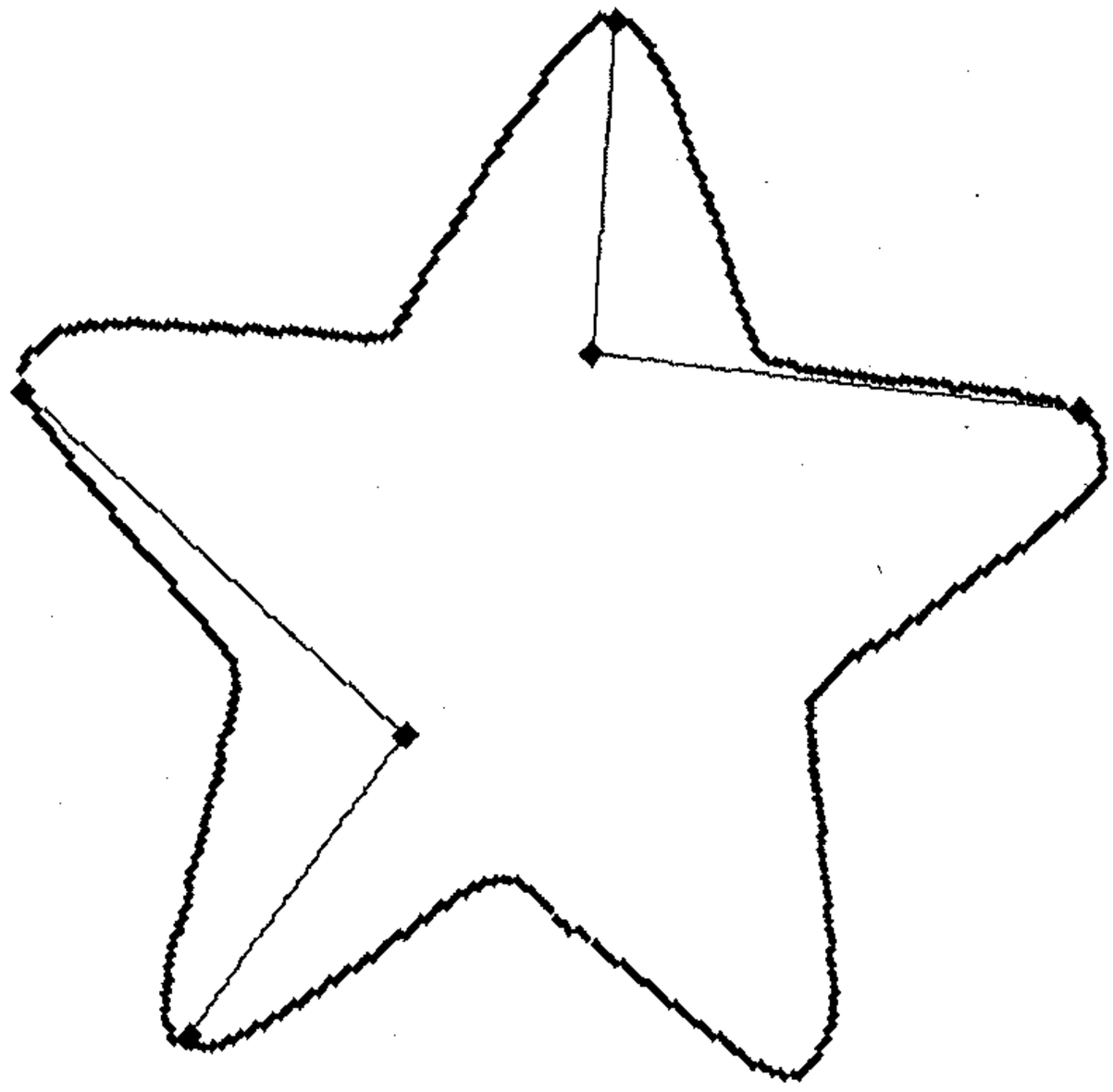

Figure 4.17: Back-projected edge image of the object of figure 4.16.

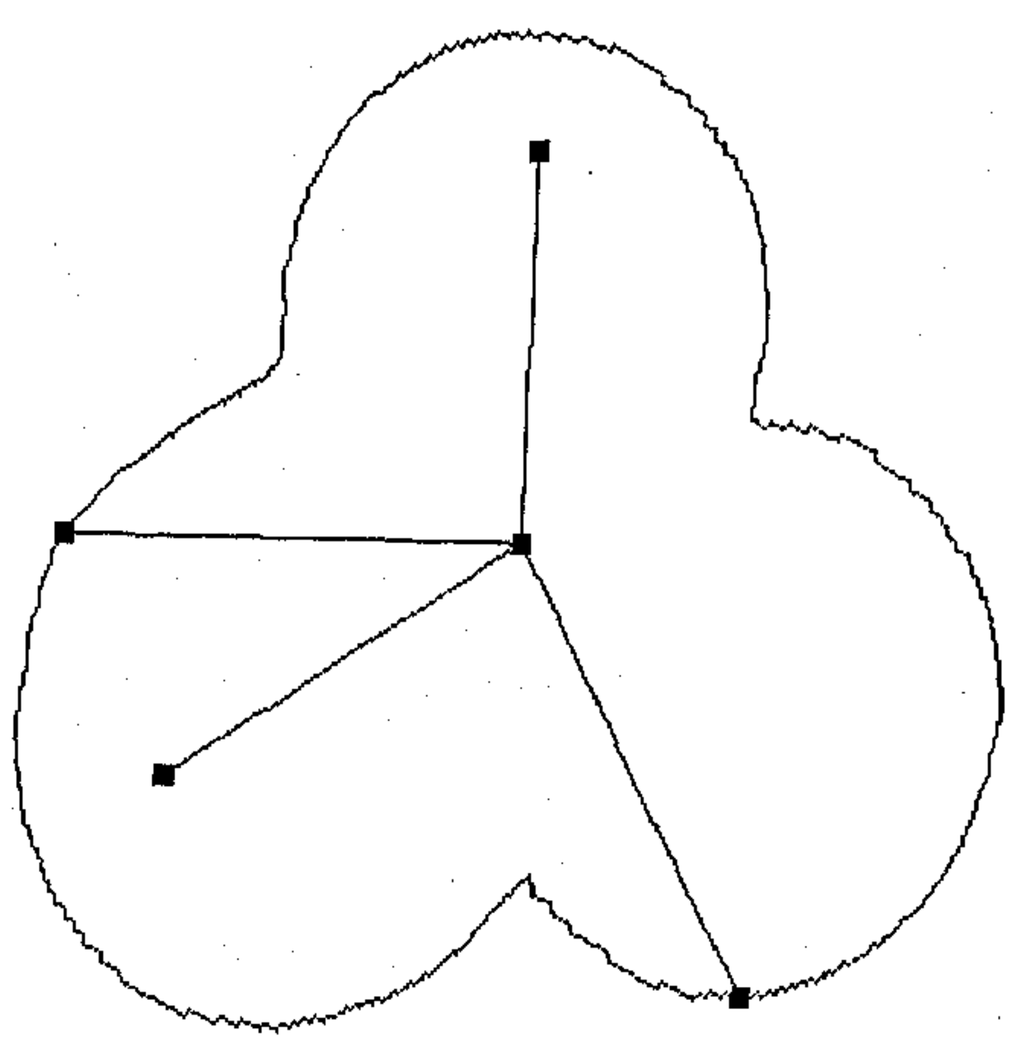

Figure 4.18: The edge image of figure 4.1 is back-projected removing both perspective and affine distortion. Compare this with the back-projection achieved in figure 4.2 where only affine distortion is removed.

Figure 4.19: Projective scene of multiple objects with skewed symmetry axes.

Table 4.5: Angles between the symmetry axes and correspondence directions before and after unskewing for multiple objects of figure 4.19.

Objects	Hex spanner	Opener	Cycle spanner	Spanner (small)	Spanner (big)
Initial Angle	83.7	86.3	97.1	92.4	81.2
Final Angle	90.0	90.4	89.6	90.1	90.2

dence directions ($\mathbf{e}_1$) of the multiple symmetries. Given, the equation of a line $\mathbf{l} = (l_1, l_2, l_3)^t$ passing through the point $\mathbf{x} = (x_1, x_2, x_3)$, $\mathbf{l}.\mathbf{x} = 0$, the problem is to find the minimum of $||\mathbf{L}\mathbf{x}||^2$ subject to $||\mathbf{x}|| = 1$, where L is $n \times 3$ matrix, n being the number of symmetries. The solution is the eigenvector of $\mathbf{L}^t\mathbf{L}$ with least eigenvalue. Similar linear least square technique is used for equation (4.7) to estimate $\mathbf{T}_a$ for multiple symmetries following chapter 3. Methodologies minimizing non-linear functions, as detailed in chapter 3, could as well be used.

Results of applying this least squares estimator to the projective scene of figure 4.19 are given in table 4.5 and figure 4.20.

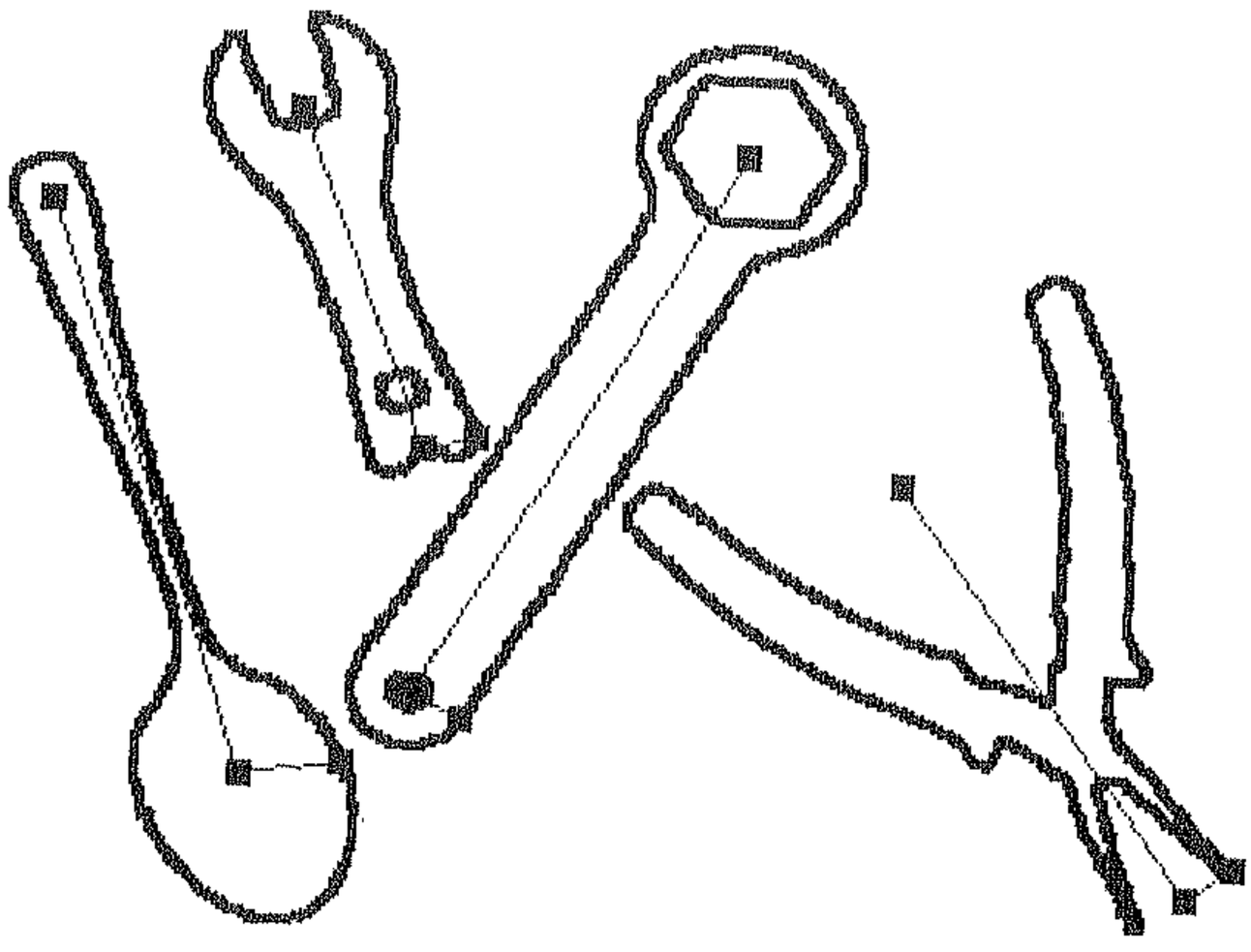

Figure 4.20: Back-projected image of multiple objects of figure 4.19.

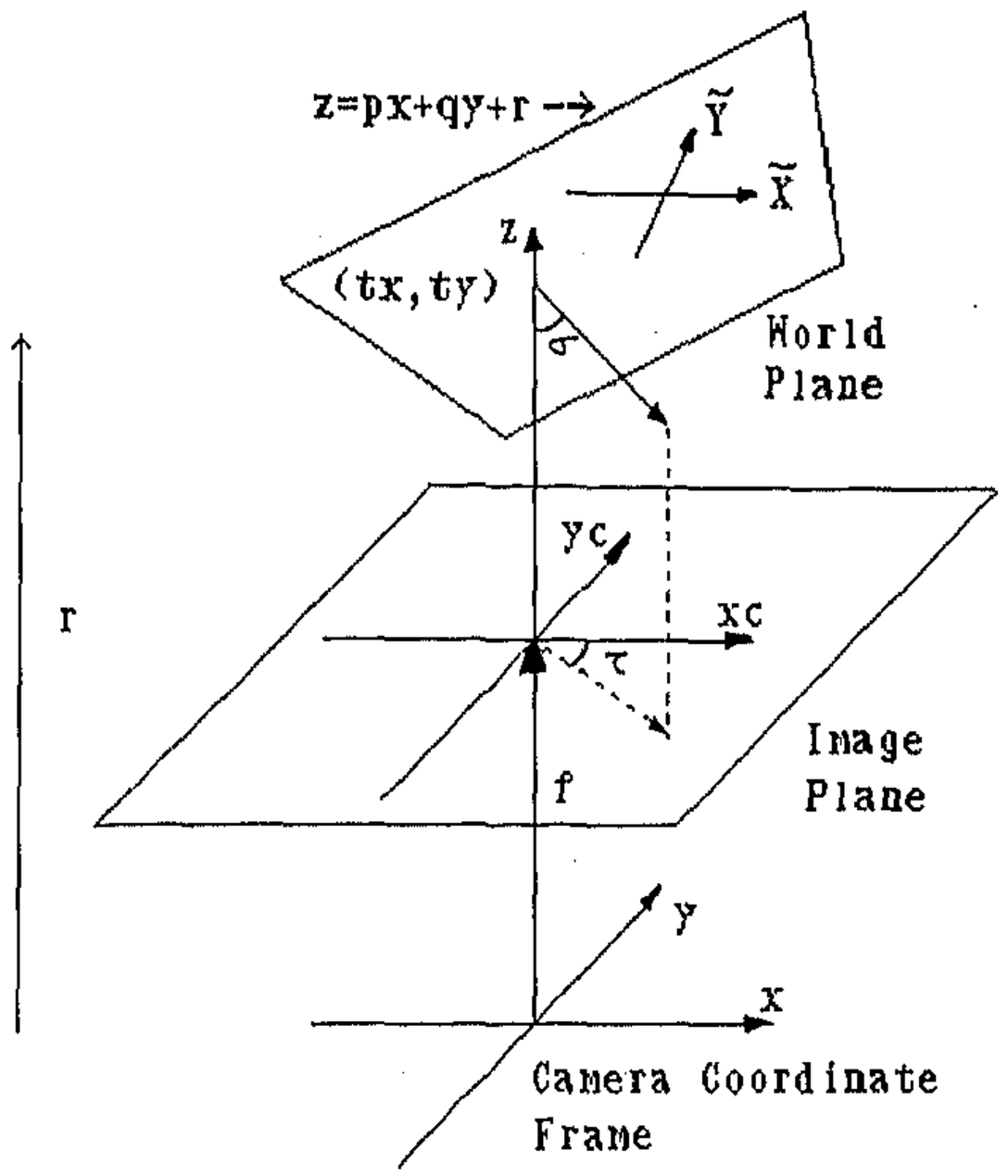

Figure 4.21: Camera and world plane geometry. Slant (σ) and tilt (τ) angles of the world plane are shown with respect to the camera co-ordinate frame.

4.4.2 Slant and tilt determination

If the camera aspect ratio is known, so that camera projection is scaled orthography, then slant and tilt of the object plane is recovered for the affine camera in section 3.4.2 [52]. Here, we determine parameters for camera calibration from back-projection matrix U. Refer to figure 4.21. The world plane is expressed in camera coordinates as

$$z = px + qy + r \tag{4.13}$$

Following [28] and the notations followed by them, the induced frame $\tilde{X}$ and $\tilde{Y}$ axes in the world plane has its centre (0, 0, r) and its $\tilde{X}$ axis in the same plane as of camera x axis. Hence the induced frame position and orientation depend only on p, q and r. We assume that camera is calibrated and camera parameters like focal length and optical centre are known. Further a rotation, (say) θ and translation,

(say) t_x and t_y, within the plane maps the induced coordinate frame onto the world frame. This means that the projective transformation is represented by the six parameters p, q, r, θ, t_x and t_y. Therefore, the back-projection U is given by [63],

$$\mathtt{U} = \mathtt{M}(p,\ q)\ .\ \mathtt{H}(r,\ \theta,\ t_x,\ t_y) \tag{4.14}$$

where $\mathtt{M}(p,\ q)$ is the central projection matrix and $\mathtt{H}(r,\ \theta,\ t_x,\ t_y)$ is an equiform matrix. Following [63], we write

$$\mathtt{M} = \begin{bmatrix} \frac{1}{n} & \frac{-pq}{mn} & 0 \\ 0 & \frac{n}{m} & 0 \\ \frac{-p}{n} & \frac{-q}{mn} & 1 \end{bmatrix} \tag{4.15}$$

and

$$\mathtt{H} = \begin{bmatrix} c & -s & st_y - ct_x \\ s & c & -ct_y - st_x \\ 0 & 0 & r \end{bmatrix} \tag{4.16}$$

where, $n = \sqrt{1+p^2}$, $m = \sqrt{1+p^2+q^2}$, $c = \cos\theta$ and $s = \sin\theta$. The back-projection matrix U is given by,

$$\mathtt{U} = k \begin{bmatrix} (mc - pqs) & -(ms + pqc) & ([st_y - ct_x]m + ct_y pq + st_x pq) \\ sn^2 & cn^2 & -(ct_y + st_x)n^2 \\ -(mpc + sq) & (mps - qc) & ([ct_x - st_y]mp + [ct_y + st_x]q + mnr) \end{bmatrix} \tag{4.17}$$

Comparing equation (4.17) with

$$\mathtt{U} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix} \tag{4.18}$$

where U is evaluated following equation (4.9). We find that:

$$\frac{D^2 + E^2}{DG + EH} = -\frac{1+p^2}{q} \tag{4.19}$$

and

$$\left(\frac{EG - DH}{DG + EH}\right)^2 = \frac{(1+p^2+q^2)p^2}{q^2} \tag{4.20}$$

Eliminating q from equations (4.19) and (4.20) gives a cubic in p^2 with only one positive real root, which we can solve apart from the sign ambiguity in p. Substituting the value of p in equation (4.19), solves for q.

Table 4.6: Slant and tilt values for the projective camera.

Image	Symmetry		Circle		Measured
	σ	τ	σ	τ	σ
figure 4.14	48	78	51	87	52
figure 4.16	32	63	35	69	35
figure 4.19	44	70	47	74	45

Representing p and q in the spherical polar coordinates, which has several advantages and most familiar to human perception, the slant (σ) and tilt (τ) of the induced frame with respect to camera coordinate axes is given by $\sigma = \cos^{-1}(\{1+p^2+q^2\}^{-1/2})$ and $\tau = \tan^{-1}(q/p)$.

The slant (σ) and tilt (τ) angles calculated for images of figures 4.14, 4.16 and 4.19 are given in the table 4.6. The results are compared to (a) slant and tilt obtained by a method which back-projects a circle under perspective [63]; (b) approximate measurements from the camera position.

Similarly, k, r, θ and t_x, t_y can be evaluated [63] using p and q values calculated earlier:

$$\begin{aligned}
k &= +\sqrt{\frac{D^2+E^2}{(1+p^2)^2}} \\
|\mathtt{U}| &= k\,|\mathtt{M}|\,|\mathtt{H}| \\
&= kr(1+p^2+q^2)^{-1/2} \\
r &= \frac{|\mathtt{U}|}{k}(1+p^2+q^2)^{1/2} \\
\theta &= \tan^{-1}(D/E) \\
AEt_x + BEt_y &= -CE \\
BDt_x + BEt_y &= -BF
\end{aligned}$$

4.4.3 Planarity tests

Suppose two symmetric planar objects are not co-planar; can this be detected from the image? If the objects are not co-planar *then* equation (4.5) may not have a

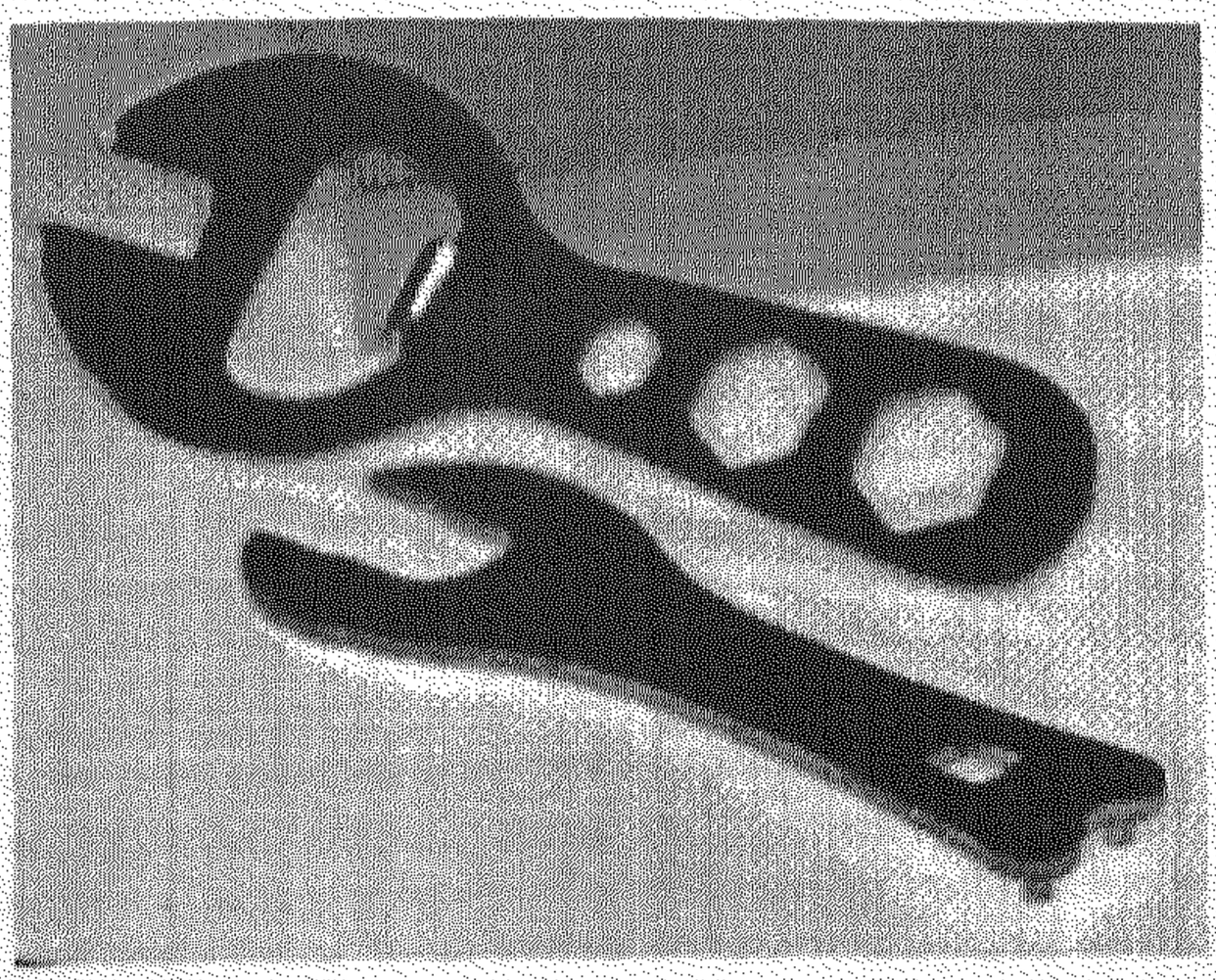

Figure 4.22: Projective image of two non-co-planar objects.

solution. In addition, $(\mathbf{e}_1^a \times \mathbf{e}_1^b)$ of equation (4.3) and the transformation T_p which transform line l' to l_∞ of equation (4.4) may not exist for two non-co-planar symmetries a and b.

The planarity tests derived in section 3.4.3 for affine camera, also hold equally good for this case. The constraints evaluated for λ (λ^2 is the ratio of eigenvalues of V (see section 3.2.2)) to be real and positive and V to be positive definite i.e. $\alpha\gamma > \beta^2$ or $det\ V \geq 0$, define a region for acceptable $\{\alpha : \beta : \gamma\}$ solutions. If solution does not lie in this region then the image cannot have arisen from co-planar objects. These tests will always be passed if the objects *are* co-planar (subject to image noise), but non-co-planar objects may fail. The unskewed guiding vectors may not be orthogonal to each other for non-co-planar objects.

To demonstrate the argument, we have taken the projective image of the figure 4.22 where two spanners are in different plane. The calculated values of $\{\alpha : \beta : \gamma\}$ is $\{-.081 : -.643 : 1\}$ which proves that the objects are non-co-planar.

4.5 Discussions

We have shown applications of image constraints due to reflectional symmetry in chapter 3. Here we have extended the same to a more general form of transformation namely projective transformation. In particular, bilateral symmetry restricts the projective transformation between corresponding image contours to a four dimensional subset of the planar projective group. This constraint allows these contours to be discriminated from other projectivity related image curves. We discuss issues relating extending these ideas to 3D objects and for other object relations in chapter 7.

At this point, we may claim that object images up to similarity transform could be recovered even from the "skewed" scene due to *non-fronto-parallel* viewing. This is achieved without utilising any knowledge of the camera parameters or the view point information and only using local symmetry information, which are in abundance both in natural or man-made objects. Having said this, we present an *almost* real time implementation of an existing symmetry detection algorithm for planar shapes under similarity transform in the next chapter.

Chapter 5

Symmetry Analysis using Wave Propagation

5.1 Introduction

In recent years, the wave diffusion algorithm [65] has attracted considerable attention because of its potential to evaluate *almost* all the *perceptually relevant* symmetry set of a 2D shape. However, in chapter 1, we have noted that its drawback is the slow execution time caused largely due to the diffusion step. In the next section, we recall the theoretical basis of the wave process and show why the diffusion step is, indeed, necessary.

Originally, Blum [9] suggested the analogy of a prairie fire ("Grassfire") as the basis for a perceptually plausible parallel algorithm for computing the SAT [9]. Various numerical algorithms have been developed to implement the "grassfire" on discrete images (e.g. Rosenfeld and Pflatz [61], Mott-Smith [47] and Montanari [46]). Given an area A of dry grassland, if the fire starts at the edge of A and if all factors other than the shape of A are excluded, then the propagating front of the fire contains important information about the shape A. The correspondence between A and the set of quench points S together with the function q expressing the time at which the fire reaches the quench point set S correspond to the SAT of A. Unfortunately, this simple quenching model, which amounts to solving the heat or diffusion equation with initial conditions determined by the boundary of A, allows each boundary

point of A to participate in just one local symmetry, as required for the SAT. The SLS [11], *symmetry set* [32] and PISA [40] require a boundary point to participate in several local symmetries: in terms of the grassfire, advancing wave fronts have to be able to pass through each other, suggesting adoption of the wave equation instead of heat/diffusion. Scott, Turner and Zisserman [65] have implemented this wave process to detect the symmetry set. The wave fronts are also extended outside the image matrix. They have noted that without some damping on the wave front, the wave process becomes numerically unstable and so they have added a diffusion process to the wave front. This has generated the desired result but made the algorithm more cpu-intensive. Consequently, the methodology has lost its practical significance inspite of its success in symmetry detection. At this point, the reader may wish to compare figure 5.5 with figure 5.7 both of which are supposed to give symmetry sets of a hammer (after post-processing) using wave process, the later (figure 5.7) being generated *without* adding diffusion to the propagating wave front.

We present a parallel implementation of the wave diffusion algorithm on a transputer network in section 5.3. Though many workers in computer vision talk about the possibility of parallel implementation of their algorithms, few actual implementations have been reported (See [12] for a useful collection). Even fewer parallel implementations have been reported for "higher level" processes such as the computation of shape descriptions. Our algorithm has been implemented in a loosely coupled MIMD transputer network and significant speed up has been demonstrated.

It has been realized that similar result as in wave diffusion method could also be achieved by replacing the wave front as a stream of particles. This idea, which we call *normal transform*, is implemented in section 5.4. This has generated an *almost* real-time symmetry detection algorithm without the extra burden of parallel hardware. The work described in this chapter is also partially covered in [48].

5.2 Wave diffusion algorithm

The issue to be addressed is why a diffusion step is needed, since the meeting points of the two wave fronts (originating from the image contour) give the desired symmetry set points and since the addition of a diffusion process reduces the efficiency of the algorithm. The mathematics of wave generation [70] shows that a numerical implementation of the wave process and the condition that two wave fronts interact at the same place at the same time (for symmetry), depends on the progressive

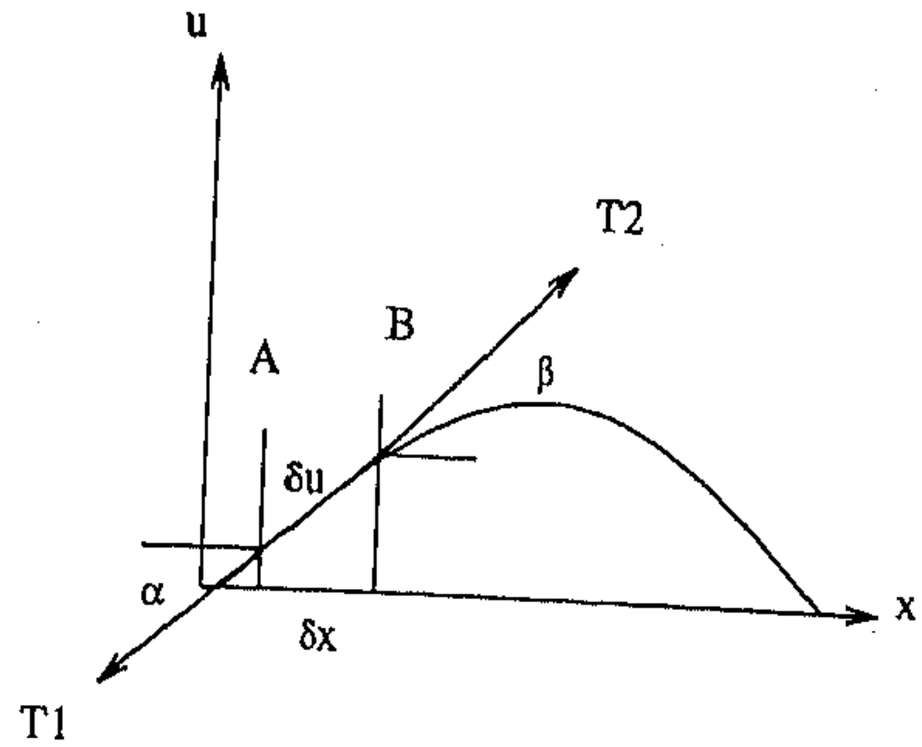

Figure 5.1: "Perfectly flexible" string for wave generation.

damping of the wave front. To see this, consider a perfectly flexible string of uniform density ρ stretched to a uniform tension between two points $x = 0$ and $x = l$ as shown in figure 5.1.

Since, by assumption, the string offers no resistance to bending, the tension is tangential to the string at each point. If T_1 and T_2 are the tensions at points A and B respectively and since all motions are assumed transverse (i.e. perpendicular to the x-axis)

$$T_1 \cos\alpha = T_2 \cos\beta = T = Constant \tag{5.1}$$

Resolving in the u-direction for the small element AB,

$$\rho \delta s \frac{\delta^2 u}{\delta t^2} = T_2 \sin\beta - T_1 \sin\alpha \tag{5.2}$$

Using equation (5.1) and approximating the length of AB, δs *by* δx, and also assuming small transverse displacement of the string,

$$\rho \frac{\delta x}{T} \frac{\delta^2 u}{\delta t^2} = \tan\beta - \tan\alpha \tag{5.3}$$

But, $\tan\alpha$ and $\tan\beta$ are the gradients of the string at points A and B respectively and are given by,

$$\tan\alpha = \left(\frac{\delta u}{\delta x}\right)_A, \ \tan\beta = \left(\frac{\delta u}{\delta x}\right)_B \tag{5.4}$$

the partial derivative being required as $u \to f(x,t)$. In the limiting case $\delta s = \delta x =$ *small distance*,

$$\begin{aligned} \tan\beta - \tan\alpha &= \left[\left(\frac{\delta u}{\delta x}\right)_A + \delta x \frac{\delta}{\delta x}\left(\frac{\delta u}{\delta x}\right) + \cdots\right] - \left(\frac{\delta u}{\delta x}\right)_A \\ &= \delta x \left(\frac{\delta^2 u}{\delta x^2}\right)_A + \textit{higher order in } \delta x \end{aligned} \tag{5.5}$$

Using equations (5.3) and (5.5) and letting $\delta x \to 0$, we can write,

$$\rho \frac{\delta^2 u}{\delta t^2} = T \frac{\delta^2 u}{\delta x^2} \tag{5.6}$$

$$\text{or} \quad \frac{\delta^2 u}{\delta x^2} = \frac{1}{c^2}\frac{\delta^2 u}{\delta t^2} \tag{5.7}$$

where, $c^2 = \frac{T}{\rho}$ and c is the wave constant. This is the wave equation for 1D which we will subsequently use for symmetry detection. This equation describes the motion of the string or simulates a wave process assuming

- the mathematical "string" has a continuous distribution of mass, and
- the displacements are small.

In contrast, in a numerical implementation of the wave process,

- the physical string is composed of large but finite number of individual particles (in our case forming an equi-spaced, quantised grid), and

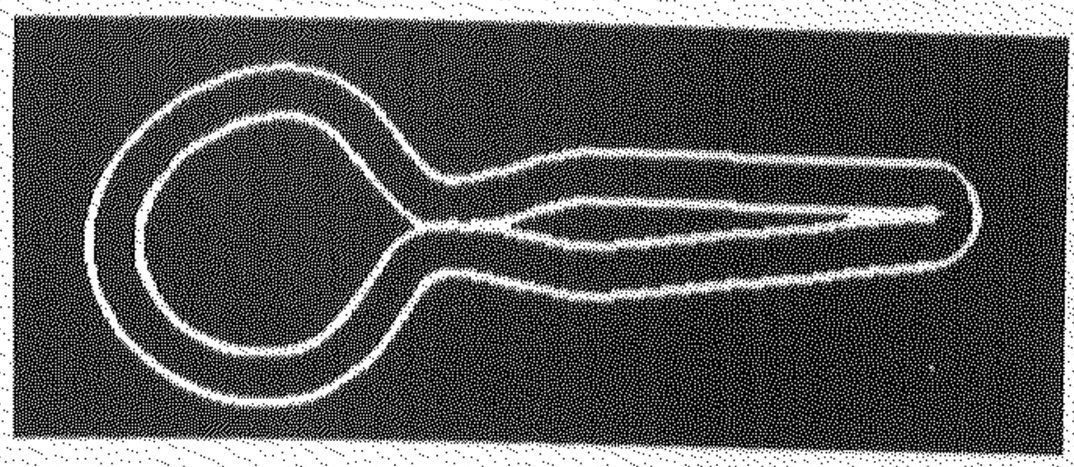

Figure 5.2: Propagation of the "shape" profile of a spoon shape.

- there is no restriction on the transverse displacements.

These lead to a non-linear partial differential equation unless some sort of restrictions are imposed on the wave process. Analyzing the properties of solutions of the wave equation, choosing initial displacements in terms of wave $u_0(x)$ and initial wave velocity zero, the solution of the wave equation can be given in the form (d'Alembert's solution)

$$u(x,t) = u_0(x+ct) + u_0(x-ct) \tag{5.8}$$

This solution reveals the important property that a flow vector travels in two opposite directions from an initial position. Therefore, the wave equation ideally propagates a "shape" vector coherently and preserves the distribution of flow fields but it does not impose any constraint on it by itself. Figure 5.2 shows how the wave profile (after gray value thresholding) of a spoon shape moves in both the directions.

A problem arises when this is implemented numerically particularly for a finite image matrix. If u'_{t+1} and u'_t are the wave velocities at the instants t and $t+1$, then

$$c^2 \frac{\delta^2 u}{\delta x^2} = \frac{\delta^2 u}{\delta t^2} \simeq u'_{t+1} - u'_t \tag{5.9}$$

$$\text{Similarly,} \quad u_{t+1} = u_t + u'_t \tag{5.10}$$

Because of the poor conditioning of second order partial derivatives and quantization errors, the shape vector loses its coherence and preservation of the distribution of the flow field. Now this instability can be reduced by choosing a low propagation speed (dictated by wave constant c). Therefore, a need arises which will impose a restriction on the wave process by lowering the propagation speed with a damping effect. The diffusion process provides such facilities. The diffusion equation is

$$\frac{\delta^2 u}{\delta x^2} = \frac{1}{\kappa^2}\frac{\delta u}{\delta t} \tag{5.11}$$

where κ is the diffusion constant. It can be shown that the propagation speed decreases according to the square root of the time elapsed [66]. The combined wave and diffusion process proposed in [65] has the following desirable features.

- It imposes a constraint on the wave process reducing non-linear changes or unexpected pulses.
- It reduces quantization errors reducing propagation speed.
- It also imposes a consistent smoothing on the flow field towards the desired distribution. The propagating front has a flat crest compared to a sharp peak. This results in a larger window for overlapping/meeting of two wave fronts which is the necessary and sufficient condition for symmetry point generation.
- Finally, from the property of diffusion process, it propagates a weaker influence as the distance of the flow field from the source increases. This is particularly important in the case of symmetry detection, since "weak" symmetries (due to noise etc.) should give weaker responses compared to "strong" symmetries (perceptually relevant).

In theory, the combined process will carry on indefinitely given fixed initial values. Gong [66] suggests that the speed of propagation (C_{wd}) and the time for stopping the process (T_{wd}) should be chosen according to

$$C_{wd} = c + \frac{1}{logl} \tag{5.12}$$

$$T_{wd} = \frac{l}{c + \frac{1}{logl}} \tag{5.13}$$

where l is the distance of propagation; however this is an experimental prediction rather than theoretically verified relation.

5.2.1 Implementation of wave diffusion process

Edge images are extracted from the 2D images (of real objects) using the Canny edge operator [15]. The edge pixels are set to value 1 whereas pixel values are elsewhere 0. The standard numerical implementation [65] of wave and diffusion equations is followed. Note that in case of 2D image, the wave and diffusion processes solve $u(\gamma, t)$ and the corresponding equations are

$$\begin{aligned} \nabla^2 \mathrm{u} &= \frac{1}{c^2}\frac{\delta^2 \mathrm{u}}{\delta t^2} \\ \nabla^2 \mathrm{u} &= \frac{1}{\kappa^2}\frac{\delta \mathrm{u}}{\delta t} \end{aligned} \tag{5.14}$$

respectively. Initial conditions are,

$$u(\gamma, t)|_{t=0} = f(x, y) \quad and \quad \left.\frac{\delta \mathrm{u}}{\delta t}\right|_{t=0} = 0$$

where $f(x, y)$ is the 2D (closed) edge contour. The numerical values of wave and diffusion constants are taken as $c^2 = 0.05$ and $\kappa^2 = 0.01$ respectively. Each iteration consists of the following steps.

1. Wave process
   - Compute $\nabla^2 \mathrm{u}|_{t=t_i} = \frac{\delta^2 \mathrm{u}}{\delta x^2} + \frac{\delta^2 \mathrm{u}}{\delta y^2}$ using standard 3×3 convolution kernel.
   - Compute the acceleration $\left.\frac{\delta^2 \mathrm{u}}{\delta t^2}\right|_{t=t_i} = c^2 \nabla^2 \mathrm{u}|_{t=t_i}$.

- Compute the velocity $\left.\frac{\delta \mathbf{u}}{\delta t}\right|_{t=t_i+1/2} = \left.\frac{\delta \mathbf{u}}{\delta t}\right|_{t=t_i} + \left.\frac{\delta^2 \mathbf{u}}{\delta t^2}\right|_{t=t_i}$.
- Compute the displacement $\mathbf{u}|_{t=t_i+1/2} = \mathbf{u}|_{t=t_i} + \left.\frac{\delta \mathbf{u}}{\delta t}\right|_{t=t_i+1/2}$.

2. Diffusion process
   - Compute $\nabla^2 \mathbf{u}|_{t=t_i+1/2}$ *and* $\nabla^2 \mathbf{u}'|_{t=t_i+1/2}$.
   - Compute the displacement $\mathbf{u}|_{t=t_i+1} = \mathbf{u}|_{t=t_i+1/2} + \kappa^2 \nabla^2 \mathbf{u}|_{t=t_i+1/2}$ and the velocity $\left.\frac{\delta \mathbf{u}}{\delta t}\right|_{t=t_i+1} = \left.\frac{\delta \mathbf{u}}{\delta t}\right|_{t=t_i+1/2} + \kappa^2 \nabla^2 \mathbf{u}'|_{t=t_i+1/2}$.
3. Store accumulated energy $[u(\gamma, t)]^2$ into accumulated energy store.
4. Update and record energy maxima at the corresponding iteration number.

This combined wave diffusion process is to be executed for a number of iterations determined by the user. The output image with accumulated energy gives the symmetry set of the image. Note that the local symmetries will show up at early iterations whereas global/reflectional symmetries, in general, will be evident at the later stage.

The execution time necessary for each iteration can be estimated from the complexity analysis of the algorithm. For a $n \times n$ image with a 3×3 Laplacian mask size, convolution computations will be of $O(n^2)$ for the entire image which (convolution) is again performed three times for every iteration. Even ignoring others, finally to record accumulated energy, $O(n^2)$ evaluation of $[u(\gamma, t)]^2$ is required for each iteration. Therefore, even though the combined wave diffusion process computes the symmetry set, the method loses its practical significance unless a faster version of the algorithm is realizable. We have implemented the algorithm in a transputer network obtaining a significant speed up as detailed in the next section. The results of implementing the algorithm on transputer network is presented in the next section.

5.3 Parallel implementation of wave diffusion algorithm

The basic problems to be taken care of in mapping an algorithm onto a loosely coupled coarse grained network of processors such as transputers are

- to keep inter-processor communication to a minimum,
- to maximally balance individual processor loads, and
- lack of parallelism due to task precedence.

To balance loads, the tasks must be distributed evenly and thus processor communication increases. Therefore, we have a trade-off between communication overhead and processor loading. To find the task precedence or the *granularity* of tasks, we analyse the data dependency, or sequential threads, in the algorithm. In the case of the wave diffusion algorithm,

- there is no global spatial dependency. Note that the generation of a symmetry axis depends on the meeting of two wave fronts. But, since a local (pixel based) convolution approach is taken, the process does not need to know the positions of the image edges a priori.
- there is intra-iteration dependency. As noted in subsection 5.2.1, in each iteration, the image is subjected to a wave process followed by a diffusion process, then finally calculation of energy maxima. This sequential order inside an iteration cannot be altered (See [65]).
- there is local dependency since each pixel is dependent on immediate neighboring pixels (8 neighbors for 3×3 convolution mask) as a result of the convolution operation.

Therefore, in distributing the wave diffusion process over a transputer network, if step (2) above is distributed, the whole image matrix has to be communicated in the processor network in every iteration due to intra-iteration sequential thread. In contrast, only a row of pixels has to be communicated (due to 3×3 convolution) across the network for every iteration if the image is divided into parts and distributed beforehand. Therefore the architecture of the network is

- the root transputer communicates to all the processors the number of iterations selected by user.
- the root transputer divides and communicates the parts of the image to individual processors.

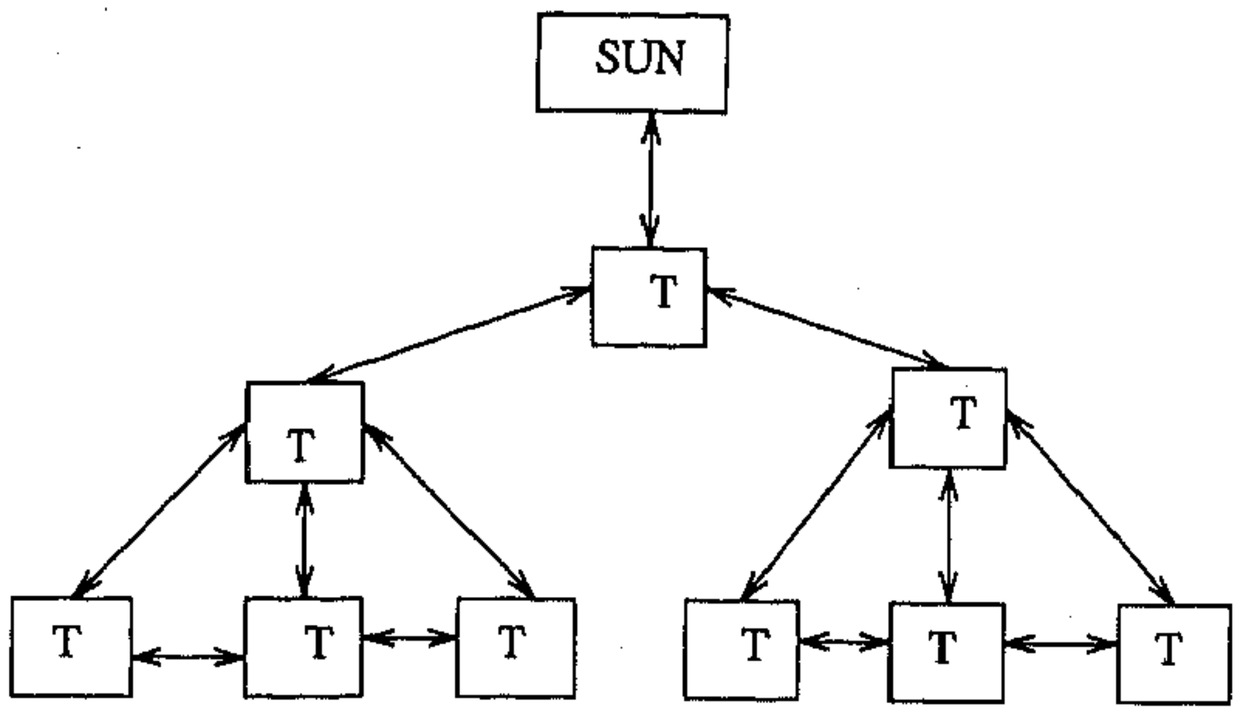

Figure 5.3: Tree architecture for distributed system for wave diffusion process.

- each worker transputer (including the root transputer) runs an identical task (step 1-4 of subsection 5.2.1).
- each task contains two threads - a "wave thread" performing the wave/ diffusion process and a "communication thread" which sends/ receives overlapped rows for convolution during every iteration. Note that images are sliced along the width (or height whichever is less) only instead of both width and height to reduce the communication overhead.

Use of threads introduces àn additional limited amount of parallelism inside the task. Since all the transputers in the network run the wave thread for an identical number of iterations, the processors are balanced and "almost" synchronized (there will be communication delay due to multiplexing of communication thread). There may be any number of transputers but the limitation in communication is due to the number of channels (which is four) available to each transputer. The overall processor architecture is the tree structure shown in figure 5.3. Note that the nodes (except those that have no child) additionally have a multiplexer thread running to receive message from an arbitrary number of input ports and to pass them through a single output port.

Postprocessing

This is a crucial step in analyzing symmetry from the images subjected to wave diffusion algorithm. The perceptually relevant symmetry set will show up only after proper postprocessing. We have found the following steps to be satisfactory though we must mention that the parameters influencing symmetry localization are image dependent.

1. Thresholding the wave/diffused image based on gray values. Note that we are not converting the image to a binary function; instead gray values less than the threshold are made zero.

2. Thresholding the processed image based on pixel density. Stray pixels due to image noise or *jagged edges* are eliminated.

3. The next step is *non-maximal suppression*. Note that, the gray value distribution in wave/diffused image will have a flat crest compared to a sharp peak. In a 3×3 window, we select the central candidate pixel if it has gray value more than at least five of its neighboring pixels. This turns out to be satisfactory for all the images we have worked with.

In addition to this, we have implemented a simple line linking procedure based on best first technique. In case of a break in symmetry line, the pixel with gray value closest to the current pixel value in a 3×3 window is marked as a potential candidate.

The parallelization of postprocessing stage is quite straightforward. Root transputer will broadcast the threshold value for pixels, pixel density over a user selected window and the parameter for non-maximal suppression. Since, postprocessing is done only once after the requisite number of iterations for wave diffusion process, there is no communication overhead due to overlapping rows (or columns) of the image window.

Results

INMOS T800 series transputers are used each delivering 15 MIPS CPU and 2.4 Mbytes/sec data communication rate. Note that the communication delay at this

Table 5.1: Comparison of processing time (in sec.) of real images subjected to wave diffusion process.

Image	Size	Iterations	Seqn. time	2Transputer	3Transputer	4Transputer	6Transputer
Hammer	172×442	60	140	119	108	92	81
Spanner	142×382	60	101	88	74	68	64
Plier	382×362	65	288	196	181	168	140

high speed of data transfer between two transputers should be almost negligible for a single row of an image. But, as each image is subjected to a minimum of 60 iterations (depending on image size), and since at each iteration a minimum of six times a row (or column) has to be communicated (two overlapped rows for each convolution), the total communication time is not insignificant. Moreover, slight communication delay during the initial iterations has increasing effect as the iteration number increases. These can be observed by the difference in processing time of (say) initial 10 iterations and 10 iterations at a later stage.

Figure 5.4 shows the application of wave diffusion algorithm on the edge image of plier (as detailed in subsection 5.2.1) implemented on the transputer network. Figure 5.5 is the snapshot of energy value of the edge image of hammer after 60 iterations of wave diffusion process. Figure 5.6 is the symmetry set of the hammer after postprocessing the image in figure 5.5. Figure 5.7 shows the result of wave process on the edge image of hammer without incorporating diffusion process in the *wave thread*. This proves the importance of diffusion process in the symmetry detection algorithm. Figure 5.8 is the symmetry set of spanner. Note that, though the rotationally symmetric points are somewhat diffused, partially because of post-processing, all the perceptually significant symmetry segments could be obtained using wave diffusion algorithm.

The total processing time in seconds (which includes CPU time, communication time and the transputer I/O time) for the image against the number of transputers used is given in table 5.1. The speed up factor (we define as the ratio of total processing time with a single processor to that of a multiprocessor) is shown in the graph in figure 5.9. It is noted that the speed up is lessened as the size of the image increases. Processing time of the sequential implementation is evaluated on the SUN SPARC station and for consistency the same C-code is used for both the cases except the additional communication protocol and configuration files necessary for parallel C.

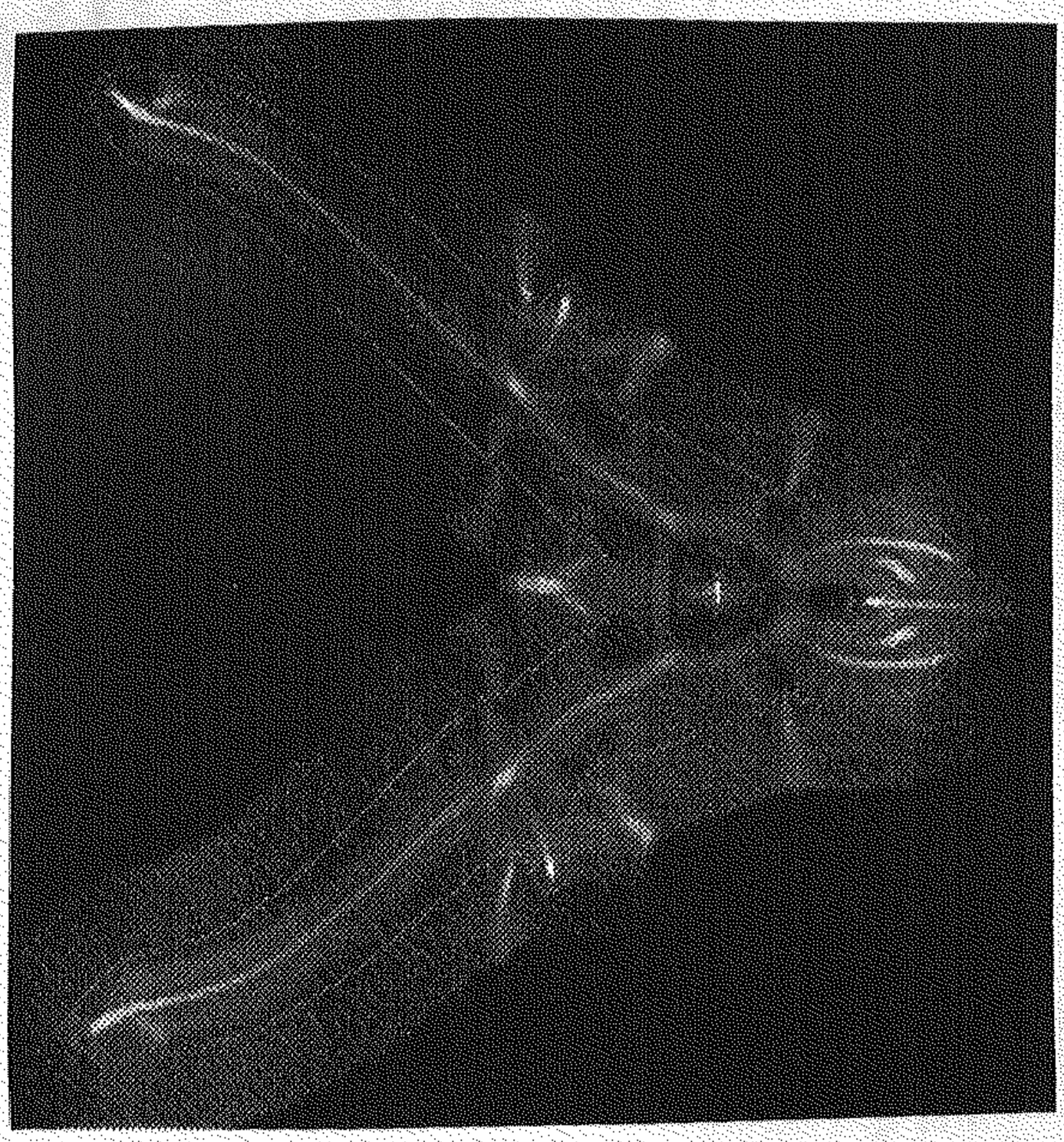

Figure 5.4: Symmetry set of the edge image of plier after 65 iterations. All perceptually relevant symmetries are present.

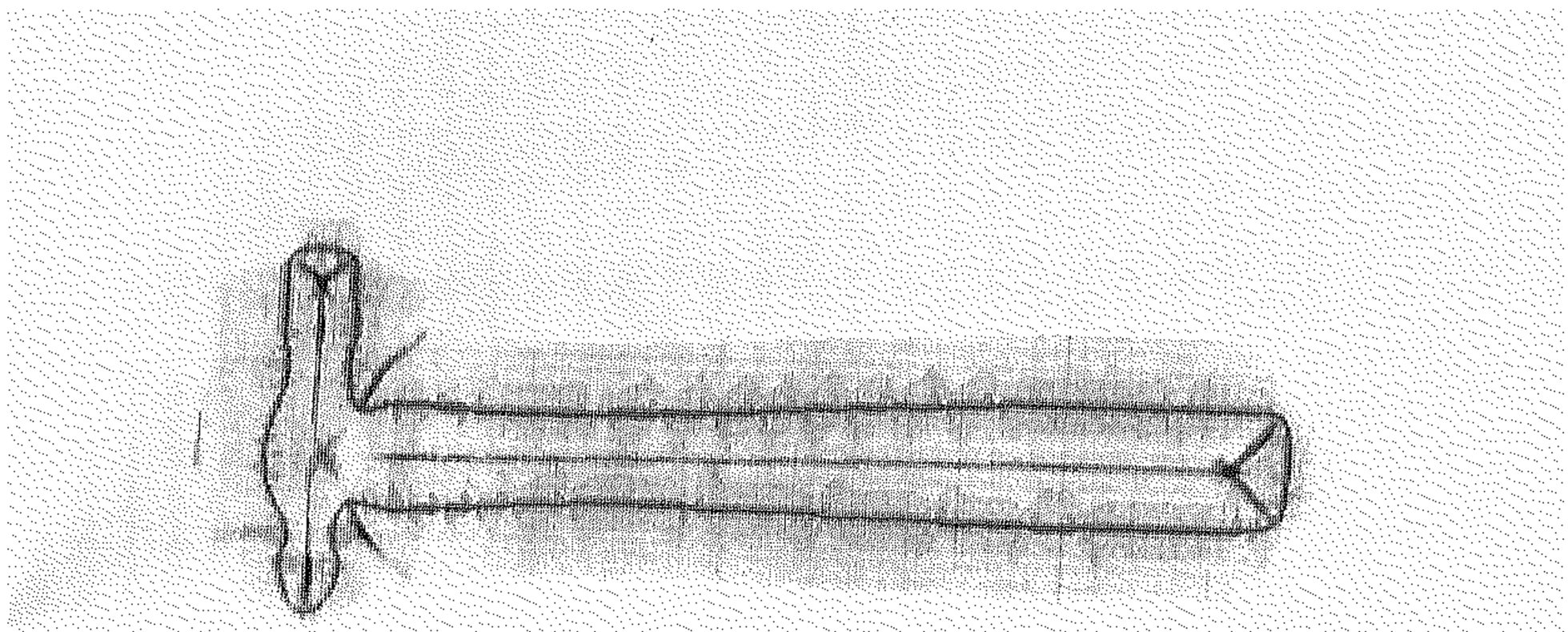

Figure 5.5: Snapshot of energy value of the edge image of hammer after 60 iterations of wave diffusion process. The edge contour has shown up as maximum energy points because of our assumption that starting function has all edge points at value 1, with 0 elsewhere.

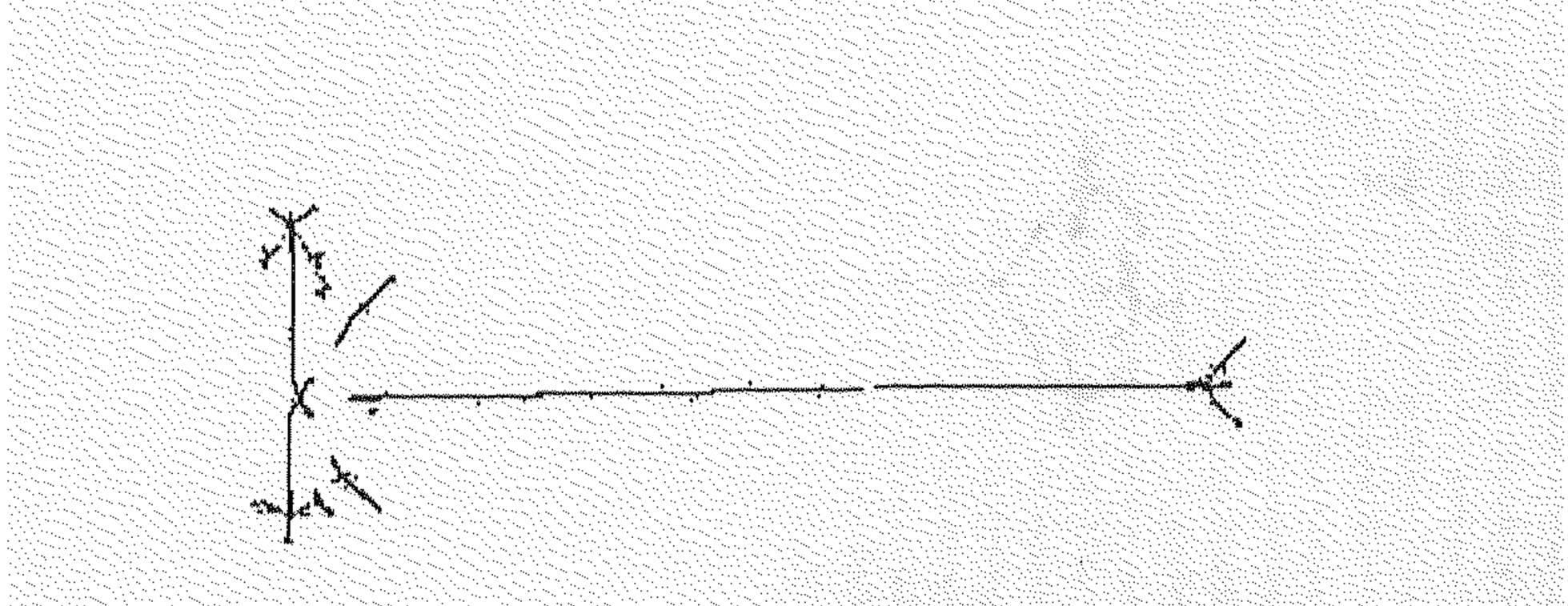

Figure 5.6: Symmetry set of hammer after post processing of the image in figure 5.5. The edge map of hammer is subtracted.

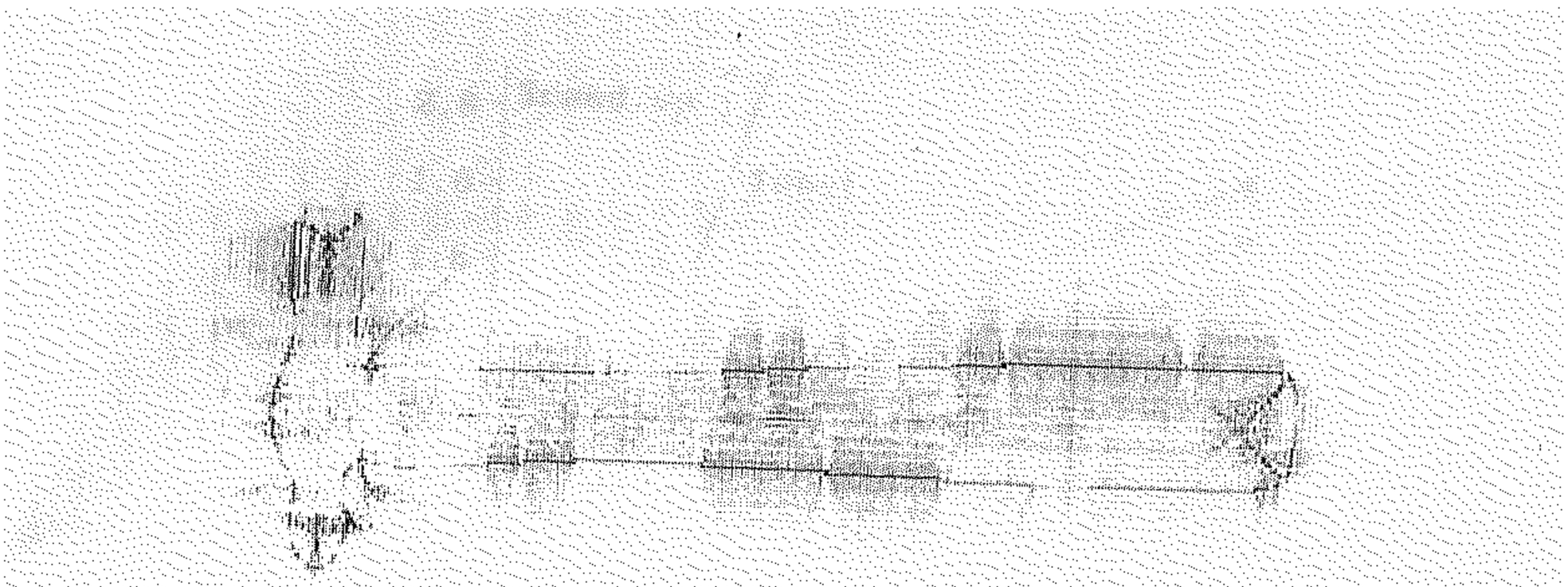

Figure 5.7: Edge image of hammer subjected to wave process without diffusion for 60 iterations.

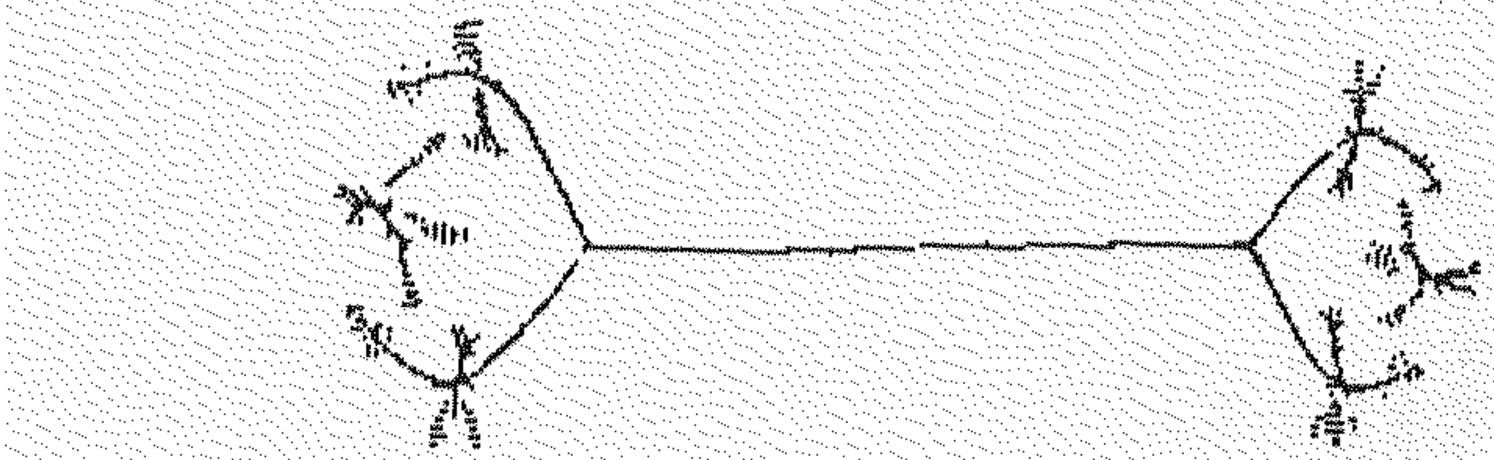

Figure 5.8: Symmetry set of the edge image of spanner after 60 iterations. Note that rotationally symmetric points are also present.

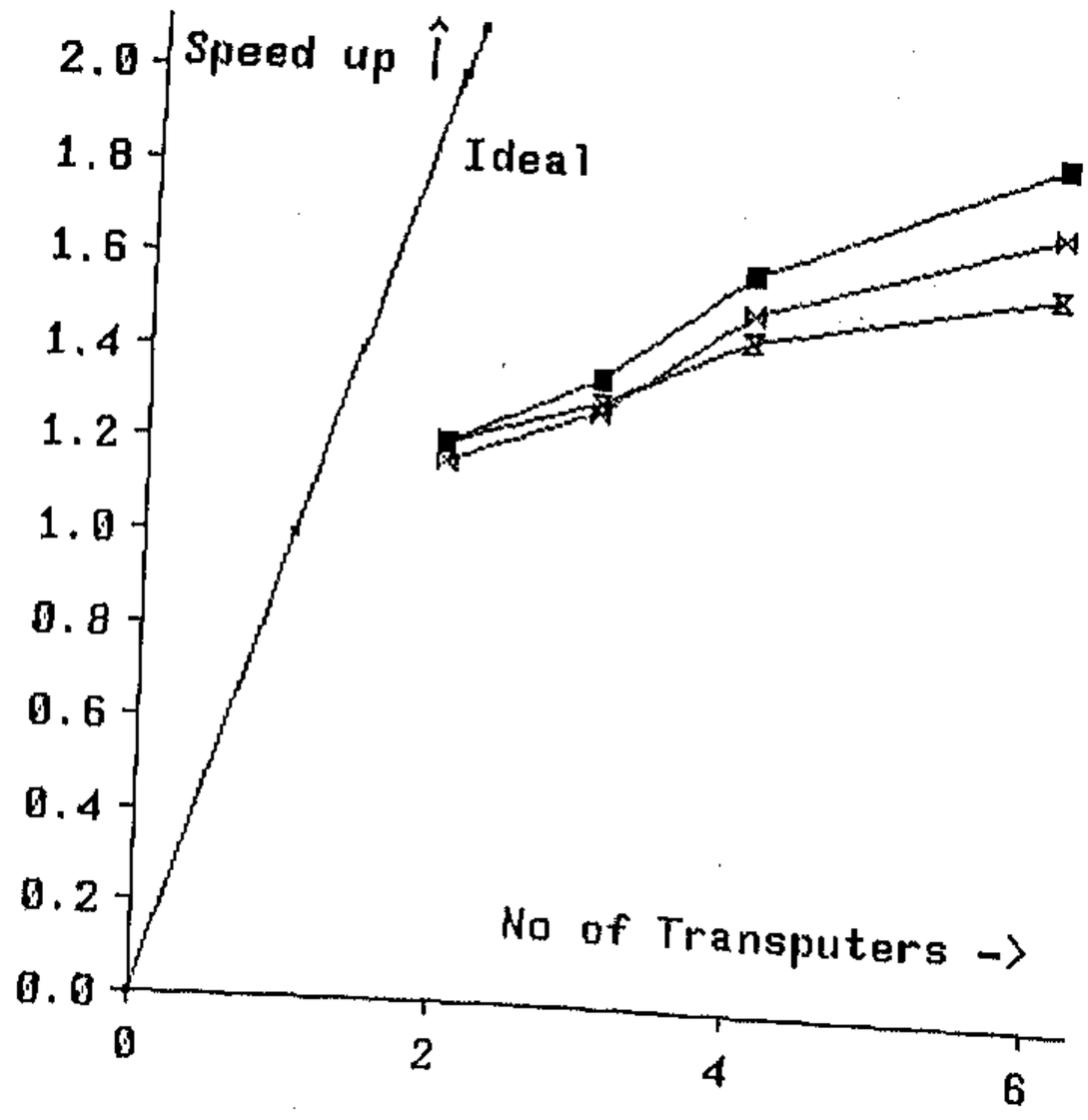

Figure 5.9: Speed up achieved in processing time against "ideal".

5.4 Normal transform

Considering the implementation of the wave diffusion process, we see that if we replace the wave front with hypothetical particles, we should get similar results provided the particle *profile* maintains the "shape" vector. The constraints, as earlier, are

- that a particle moves with constant velocity from the edge contour in both directions (as in wave propagation),
- the particles must move in a direction "normal" to the "shape" profile, and
- a hypothetical "flow through" property of particles needs be implemented numerically so that they maintain their "normal" direction even after collision.

The collision path of the particles will give the symmetry set similar to the wave diffusion process. The "flow through" property is easily implemented with an accumulator array (of image size) which will monitor the movements of particles and

will flag on the location whenever it will detect more than one particle at the same instant in a quantised grid. The algorithm is extremely easy to implement and consequently fast in execution and also easily parallelizable. But it suffers from the same problem as noted in wave diffusion process namely the digitization error in calculating normal vector at the edge point. We have implemented two different methods both of which give reasonably good results.

First, we approximate the contour with a B-spline [1]. A cautious strategy is followed by taking large number of vertex points in the approximating polygons so that almost every detail of the 2D closed contour can be replicated. If we represent the curve parametrically as follows:

$$w = \left\{ \begin{array}{c} x(t) \\ y(t) \end{array} \right| t \in (t_0, t_1) \tag{5.15}$$

The normal to the curve at $t = t_1$ is given [24] by,

$$x = x(t_1) + ty'(t_1)$$
$$y = y(t_1) - tx'(t_1)$$

where $x'(t_1)$ and $y'(t_1)$ are the values of $\frac{dx}{dt}$ and $\frac{dy}{dt}$ at $t = t_1$ respectively. The higher the number of divisions between (t_0, t_1) the more the quantization error is reduced.

The second approach approximates the contour with a collection of small straight line segments. Five consecutive pixels are approximated to a straight line using orthogonal regression and the normal is drawn from the mid-point (third point). The normal direction is determined from the eigenvalues/eigenvectors of the regression matrix of the set of five points. To take care of the quantization error (which increases with the number of iterations), the accumulator array (which keeps track of the collision path of the particles) flags a potential symmetry set point if more than one particles are detected in a pre-defined neighborhood of a quantised grid instead of a single grid point.

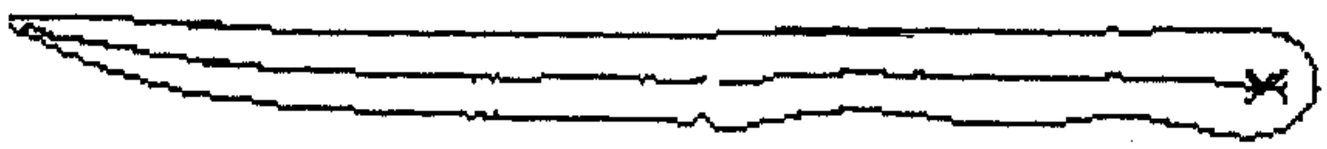

Figure 5.10: Symmetry set obtained using normal transform after fitting b-spline to edge contour of the knife.

Postprocessing

To reduce the quantization error and to differentiate between noise pixels to the symmetry set points, an approach similar to relaxation labeling is implemented. Following Brady and Asada [11], given a "putative" symmetry set point,

- look for support for the likelihood that it is indeed a symmetry set point by seeking others in the same direction within some error distance from the point under investigation.
- if such points are available, then increase the confidence that this point is a symmetry point after fashion of relaxation and interpolate intermediate points. Else discard the point as a noise point.

Results

Examples of symmetry set using normal transform method for a set of mechanical tools are shown in figures 5.10 and 5.11. Both the images are subjected to line linking algorithm as detailed earlier. For these figures, the algorithm is terminated when all the particles reach the image boundary. The processing time is insignificant compared to wave diffusion process and is essentially instantaneous. Note that fairly good results are obtained even in case of "jagged" edges of chisel after line linking. The important difference of this approach in contrast to wave diffusion process is that it has a comparatively "violent" effect in the case of image noise or missing edge pixels where as this effect will be much reduced by the smoothing action of the combined wave diffusion process.

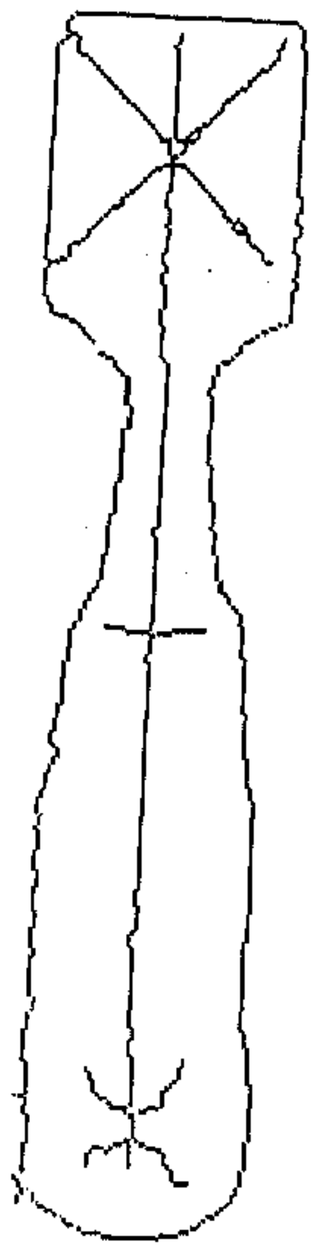

Figure 5.11: Symmetry set of a chisel with "jagged" edges after post processing as detailed in section 5.4.

5.5 Discussions

We have explained the role of the diffusion process in detecting symmetries from the mathematical basis of the wave equation and shown numerically stable result could only be obtained by using the combined effect. As this leads to an inefficient symmetry set detection algorithm, our MIMD implementation of it gives a viable alternative with considerable speed up in the processing time. Normal transform method is equally attractive and has given comparable result with respect to parallel wave diffusion implementation. We explore alternatives of MIMD parallelization of wave diffusion algorithm in chapter 7. Once the *practicality* of these symmetry detection methods are established, symbolic description of shape based on these techniques could as well be developed following [48].

So far, we have shown that symmetry set could be evaluated efficiently for planar shapes under projective, affine or similarity transform exploiting geometric properties of shape or emulating simple physical phenomenon. In the next chapter, we show an application of this symmetry set for solving a model based object recognition problem.

Chapter 6

Symmetry Set Matching using Simulated Annealing

6.1 Introduction

Shape matching is a classical problem addressed by computer vision researchers over the years. In this chapter, we present the methodology to find the best match between the symmetry set of a planar shape with that of a candidate model of the model base. Symmetry sets are generated using the parallel wave diffusion algorithm described in chapter 5. As noted earlier, because of the damping in wave front, we expect shape representation using this technique to be more noise-robust compared to other numerical techniques of symmetry generation.

We have an *integrated model graph* [54] combining symmetry sets of all the models in a given model base. The obvious advantage of building a model base using symmetry is reduction in data storage and consequently reduction in search space during matching. In case of globally symmetric shape, only a half of symmetry set is enough for shape matching. Since, only the *perceptually relevant* information are stored in the model base, the symmetry points have a better discriminatory power.

The matching constraints between the symmetry set of the image and the *integrated model graph* combining all the models are expressed as an energy function. Simulated annealing technique is used to minimize the energy function to find the

best match. The method of simulated annealing [38] is a technique that has attracted significant attention as suitable for optimization problems of large scale, especially ones where a desired global extremum is hidden among many, poorer, local extremum.

In the next section, we detail steps of feature extraction and model building given all the *perceptually relevant* symmetry sets. The energy function with the constraints of matching of symmetry sets and its minimization using simulated annealing are described in section 6.3. The experiment along with the result is given in section 6.4. Note that the symmetry generation using wave diffusion algorithm is also a function of time of propagation of wave front, variation of which results in variation of feature set parameters. However, in section 6.4, we show the key result that even for variation in the time of wave propagation, our minimization technique gives a perfect match. This is followed by discussion.

6.2 Feature extraction

To state the criterion for shape description,

- the algorithm should take care up to similarity transform of the shape particularly, rotation and scaling.
- It should be able to distinguish between shapes which are different even at a coarse scale of description. The description should be reasonably stable and robust for minor changes in image such as due to noise or illumination changes etc.
- It must be fast and possible to implement in parallel hardware.

As detailed in [48], similar to the notion of canonical frame described in earlier chapters, feature extraction and shape matching constraints are evaluated in a representative frame. The symmetry sets of the image and all the candidate models are described in the representative frame which takes care shape representation up to similarity transform (up to a certain rotational (π) ambiguity, discussed below).

The steps involved to map the symmetry set of a planar shape to the representative frame are

1. detection of major axis,
2. transformation of symmetry set to the representative frame, and
3. evaluation of feature attributes of transformed symmetry set.

Detection of major axis For a region **A** in $\mathcal{R}^2$, in this case edge contour of the object subjected to wave diffusion process or normal transform, with centre of gravity $(\bar{x}, \bar{y})$, the (Rimann) integrals $\int_A (x-\bar{x})(y-\bar{y})da$, $\int_A (x-\bar{x})^2 da$ and $\int_A (y-\bar{y})^2 da$ are well- defined [41] and finite (x and y denote the co-ordinates of the points in **A**). The major axis passing through $(\bar{x}, \bar{y})$ makes an angle θ ($\theta \in [0, \pi)$ with the x-axis where,

$$\tan 2\theta = \frac{2\int_A (x-\bar{x})(y-\bar{y})da}{\int_A (x-\bar{x})^2 da - \int_A (y-\bar{y})^2 da}$$

To increase noise robustness, weighted scatter matrix may also be used to evaluate the orientation of the major axis.

The major axis direction is invariant upto similarity transform. The intersection points of the major axis (extended in opposite directions starting from the centre of gravity) with the image contour give two stable points. In case of more than two such intersection points, the point pair which is at maximum distance apart are chosen. For shapes with multiple major axes, the one with minimum angle to the x-axis (in anti-clockwise direction) is selected.

Figure 6.1 is the edge image of a DC10 aircraft whose major axis and the centre of gravity C is calculated. The intersection points, A and B, of the major axis with the edge contour are shown.

The symmetry set of the edge image of figure 6.1 after 80 iterations of wave diffusion algorithm and non-maximal suppression [48] is shown in figure 6.2.

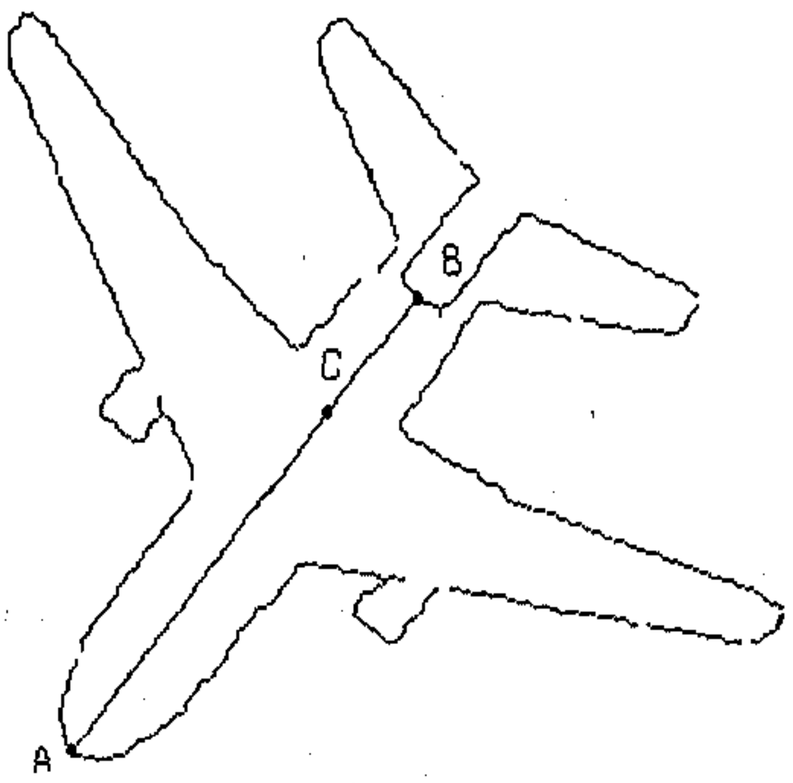

Figure 6.1: Edge image of DC10 aircraft with major axis and contour intersection points A, B and centre of gravity C.

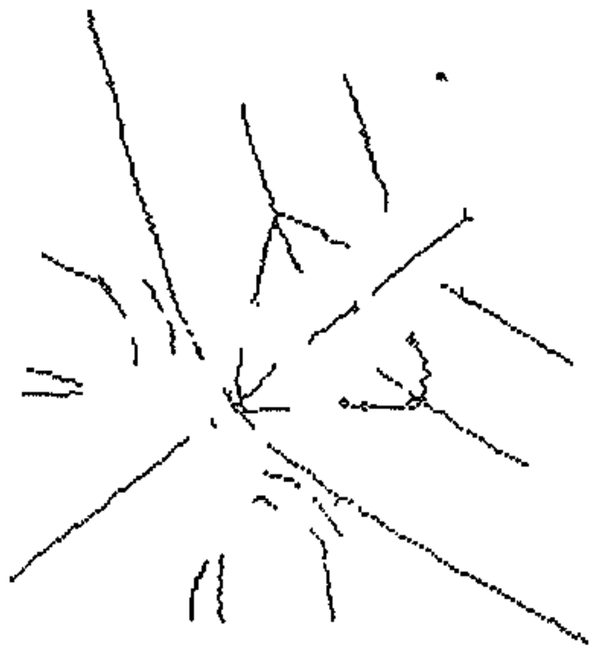

Figure 6.2: Symmetry set of the edge image of figure 6.1 after 80 iterations of wave diffusion algorithm followed by post processing.

Figure 6.3: Symmetry set of figure 6.2 in the *canonical representative frame.*

Transformation of symmetry set to the representative frame This two-point set (A, B as in figure 6.1) is transformed to a *canonical representative frame* with basis points at (1, 0) and (-1, 0) on the x-axis of the orthogonal axes system. Ignoring translation and considering rotation and scaling, the transformation is given by,

$$\mathbf{x} = \mathbf{M}\mathbf{X} \tag{6.1}$$

where $\mathbf{X}$ and $\mathbf{x}$ are two-vectors before and after mapping to the representative frame respectively and $\mathbf{M}$ is the 2×2 transformation matrix. With scaling factor λ and major axis angle (anti-clockwise, with respect to x-axis) β,

$$\mathbf{M} = \lambda \begin{bmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{bmatrix} \tag{6.2}$$

Two-point set constraints evaluate λ and β and thereby matrix $\mathbf{M}$, from equation (6.1). All the symmetry points of the 2D shape are then mapped to the *canonical representative frame* using the transformation matrix $\mathbf{M}$. The candidate models of the model base are also described in this same frame.

The symmetry set of figure 6.2, after mapping to the *canonical representative frame* using the points A, B and the basis points (1, 0) and (-1, 0), are shown in figure 6.3.

This mapping to representative frame takes care of scaling and rotation upto a certain ambiguity. Because of change in order of mapping of A, B to the basis

points, there may be a π rotation associated with each mapping. This has to be taken care of in case of feature matching when the presence of feature in the direction both (say) γ (with respect to x-axis) and $(\pi + \gamma)$ are to be checked. The symmetry axis segment(s) with orientation close (within a threshold) to the major axis (or the x-axis after transformation to the representative frame) gives the global symmetry line. For globally symmetric shape, only half of the symmetry set (say upper half of the x-axis) is required for matching. The models which have global symmetries are also stored in the same fashion.

Other not-so-robust techniques to detect global symmetry line(s) are also investigated. For a globally symmetric shape, if each symmetry segment is represented by a vector $\mathbf{S}_i$, $\sum \mathbf{S}_i$ will give the global symmetry line. We assume that the curve segments are approximated by polygon chains. In this case, the additional computing burden is to determine the direction of the symmetry vectors. The initial and final points of all symmetry segments could be obtained from the iteration numbers of the wave diffusion or normal transform method.

Evaluation of feature set Extraction and determination of feature primitives and their attributes from the symmetry set are discussed in [48]. For current implementation, we assume each symmetry axis segment to be either straight line or curved symmetry axis, fitted to a circular arc. However, extension of this methodology to axis segment approximated by cubic b-spline is quite straightforward. We ignore single bright points in the symmetry set due to rotational symmetry; though we admit its importance in case of shape representation. We present here one simple framework to represent and compare both linear and circular symmetry axis segments.

All symmetry set points are traced and the co-ordinates are recorded taking top-left corner of the image as $(0,0)$. A simple test similar to iterative endpoint fit and split arc segmentation algorithm [35] is implemented. For a segment, defined by end points $(x_1,\ y_1)$ and $(x_n,\ y_n)$, let $L = \{ \begin{pmatrix} x \\ y \end{pmatrix} \mid ax + by + c = 0\}$, where $a^2 + b^2 = 1$. For any point $(x_r,\ y_r)$, let d_r be the distance between L and $(x_r,\ y_r)$; $d_r = |ax_r + by_r + c|$. For all the points in a given axis segment if $d_r > d_{max}$, the segment is either a curved symmetry axis else a segment combining more than one linear/curve segments; the point $(x_r,\ y_r)$ is the break point. The data points of the sub-segment are approximated by a straight line. Next, we fit circle to the same set of data by minimizing

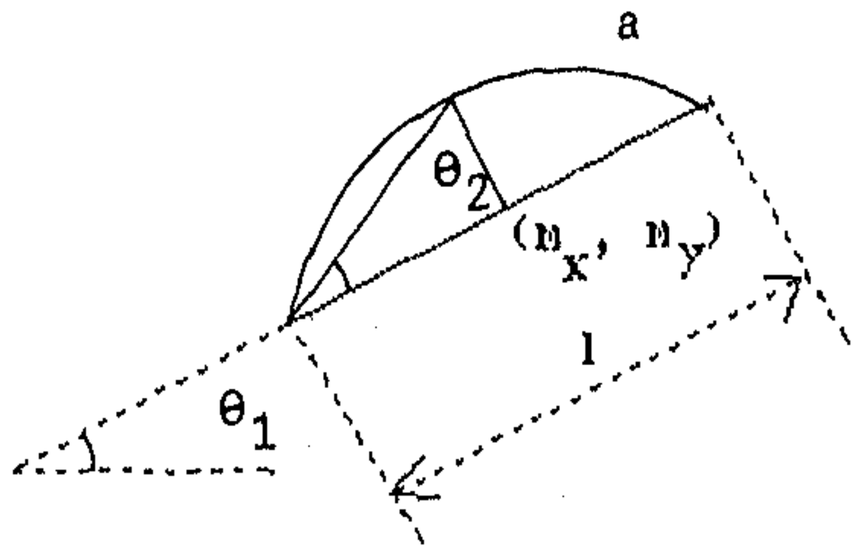

Figure 6.4: Feature attributes of a symmetry segment. The same representation scheme is used for both linear and circular segments.

$$\epsilon_c{}^2 = \sum_{i=1}^{N} (\sqrt{(x_i - a)^2 + (y_i - b)^2} - R)^2$$

where, for N set of data points, ϵ_c is the error of circle fit with centre at $(a,\ b)$ and radius R. The evaluation of circle parameters (a, b, R) are discussed in [35]. If for two consecutive circular segments, the parameters (a, b, R) are *sufficiently* close, the two segments are merged to form a single circular segment.

After describing symmetry set consisting of only linear and circular arcs, we represent symmetry segment as a feature primitive with 5-tuple attribute. Referring to figure 6.4, the attributes are $(l,\ \mathrm{m},\ (l/a),\ \theta_1,\ \theta_2)$. The Euclidean length between the end points of the symmetry segment is l, the co- ordinates of the middle point of which is m (m_x, m_y). The angles θ_1 and θ_2 are shown in figure 6.4. The parameter (l/a) is a simple measure of circularity of the feature primitive where a is the arc length of the symmetry segment. Therefore, in case of straight line, (l/a) will take a value of unity.

6.3 Shape symmetry matching

We pose the model matching problem as a relational graph isomorphism problem. Each node of the graph represents a feature primitive (linear or circular segment) with feature attributes $((l, \mathrm{m}, (l/a), \theta_1, \theta_2)$, defined above) as node attributes. The arc strength between two nodes, we define as the relational attribute between the nodes, is determined from the Euclidean distance between the mid-points (m_x, m_y) of two primitives. This distance value, within a threshold, represents relational arc strength. Thus each node may have more than one relational arc. The three stages of matching between the image and the model graph [67] are

1. structural matching of number and type of nodes,
2. relational attribute matching of arc strength, and finally
3. node attributes matching of feature primitives.

Instead of sequential processing of each of the above steps, we express the constraints of a perfect match as an energy function considering all the above aspects of graph matching together. The energy function is then minimized using simulated annealing technique.

6.3.1 Energy function for matching

Following [54], an integrated model graph is built combining feature primitives of all the models. Suppose, a network of two-dimensional binary matrix V of X rows and Y columns are constructed. X rows represent the maximum number of feature primitives in the symmetry set of an image. Y columns represent $X \times Z$ feature primitives of Z number of models, each with a maximum of X number of primitives. The matrix element V_{ijp} can take value 1 (ON state in network paradigm) when ith feature primitive of the image matches with the jth feature primitive of the pth model of the integrated model graph, otherwise $V_{ijp} = 0$ (OFF state). Note

that though V is of two dimensions, the suffix i represents row whereas both j and p represent the column dimension of V, which is a standard notation in neural network paradigm. The necessary constraints for a perfect match are as follows. The corresponding energy terms are given in the brackets.

- Each row and each column of matrix V should have one 1, (E_1).
- Total number of 1 for V should be maximum X, (E_2).
- Primitives of the image are inhibited to match with primitives of different models, (E_3).

In addition to these, the cost function due to relational (E_4) and node attribute (E_5) matching are added to the energy function. The state of V after minimizing the energy function represents a match.

The energy function [5], E, is given by,

$$E = E_1 + E_2 + E_3 + E_4 + E_5 \tag{6.3}$$

where,

$$E_1 = A/2 \sum_i \sum_{j,\, j \neq i} \sum_l \sum_p \mathrm{V}_{ilp} \mathrm{V}_{jlp} + B/2 \sum_i \sum_l \sum_{m,\, l \neq m} \sum_p \mathrm{V}_{ilp} \mathrm{V}_{imp}$$

A and B are positive constants. i, j represent row (primitives of image) and l, m (primitives of model) and p (a model of the integrated model graph) represent column of V.

$$E_2 = C/2 (\sum \sum \sum \mathrm{V}_{ilp} - X)^2$$

C is a positive constant. When two symmetry sets, each containing X primitives are perfectly matched, this term becomes zero. If for two shapes, s_1, s_2, there is a difference in number of feature primitives, X should take the minimum of X of s_1 and X of s_2.

$$E_3 = D/2 \sum_p \sum_{q,\, p \neq q} \left(\sum_i \sum_l V_{ilp}\right)\left(\sum_j \sum_m V_{jmq}\right)$$

D is a positive constant. p and q are the two different models of the integrated model graph representing column of V, the other variables being the same as E_1. As earlier, if there is a mismatch in number of primitives (X), dummy primitives are to be appended to evaluate E_3.

The cost function due to relational attribute matching is given by,

$$E_4 = F/2 \sum_i \sum_j \sum_l \sum_m \sum_p V_{ilp} V_{jmp} W_R \left| \left| C_R(i,j,p) - C_R(l,m,p) \right| \right|$$

F is a positive constant. The absolute difference of the relational attribute between ith and jth nodes of the image graph and the lth and mth nodes of the pth model of the integrated model graph is given by $|C_R(i,j,p) - C_R(l,m,p)|$. W_R is the weight or priority of the cost.

The cost function for node attribute mismatch is given by

$$E_5 = G/2 \sum_i \sum_l \sum_p V_{ilp} W_N C_N(i,l,p)$$

G is a positive constant. $C_N(i,l,p)$ is the cost due to node attribute mismatch between the ith node of image and the lth node of the pth model of the integrated model graph. For current implementation, the sum of the squares of the Euclidean distance between the 5-tuple node attributes (l, m, (l/a), θ_1, θ_2) gives the cost function. Note that weighted distance between the node attributes may also be used. W_N is the weight of the cost.

6.3.2 Optimization by simulated annealing

For a given state α of the binary network V, the probability of its state change (OFF to ON or reverse) p_i, depends on the energy gap, ΔE, between the previous and the current energy state of V. This is computed [20] using the non-linear function:

$$p_i(\alpha) = \frac{1}{1 + exp(-\Delta E_i(\alpha)/T)} \tag{6.4}$$

This follows from Boltzman probability distribution and it uses stochastic binary state to escape non-global minima during optimization. Proper selection of control parameter T (analog of temperature) and its annealing schedule influence the probability distribution during minimization. We have tested two annealing schedules, given below for the symmetry matching problem.

$$T_{next} = T_{current}(1 - T_{factor}) \tag{6.5}$$

$$T_{next} = \frac{C_t}{I} \tag{6.6}$$

In equation (6.5), new temperature T_{next} is evaluated reducing the current temperature $T_{current}$ by a constant factor (1 - T_{factor}). In equation (6.6), T_{next} is calculated dividing a constant C_t with the current iteration number I of the minimization process.

The temperature distribution for equations (6.5) and (6.6) are shown in figures 6.5 and 6.6 respectively.

6.4 Experiment

We have experimented with different sets of model base, one of which is reported here. The model base is shown in figure 6.7. The edge images and their corresponding symmetry sets, after 95 iterations of wave diffusion algorithm and post-processing as detailed in chapter 5, are in the model base. Note that all the models are described in the representative frame. Since the shapes are globally symmetric

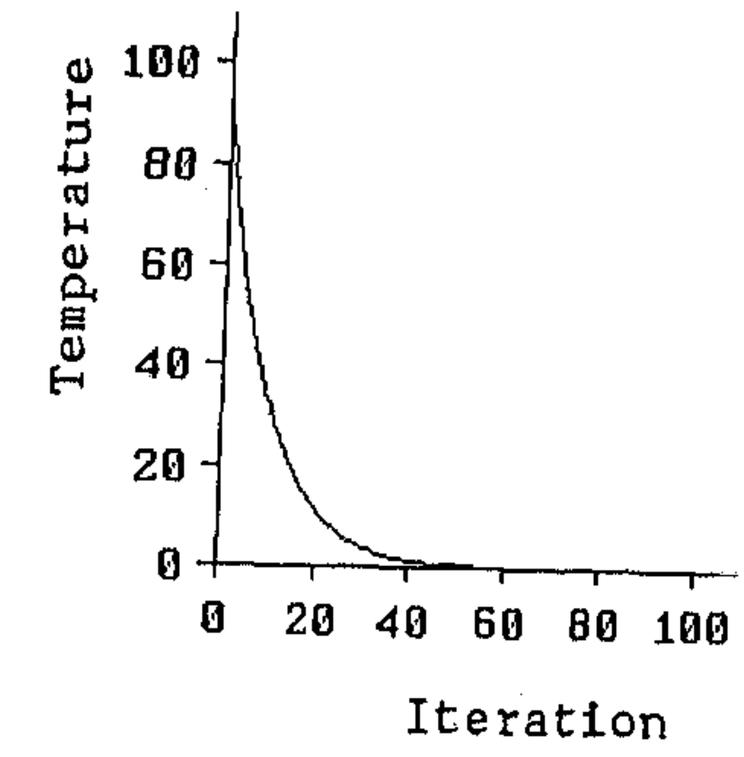

Figure 6.5: Temperature distribution for equation (6.5) up to 100 iterations. The starting temperature and T_{factor} are 100 and 0.1 respectively.

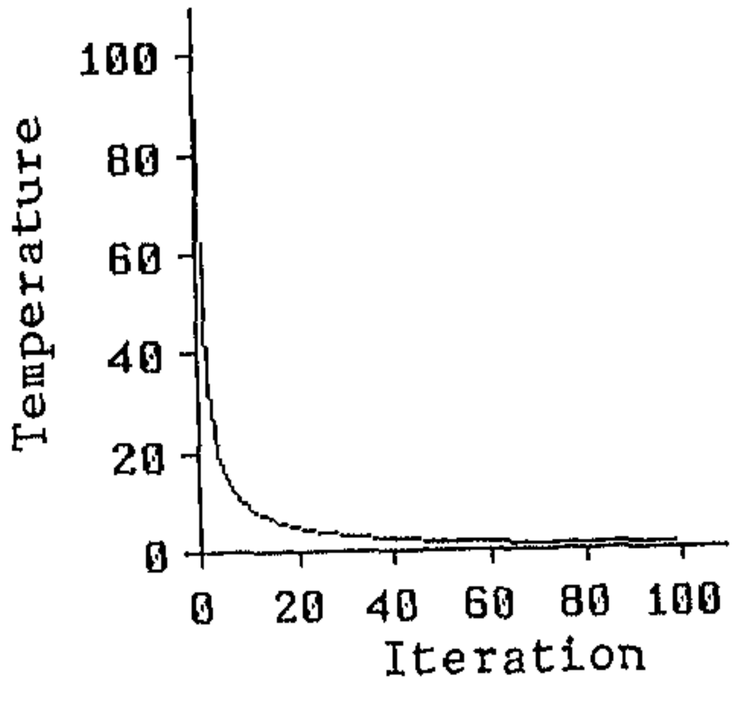

Figure 6.6: Temperature distribution following equation (6.6) for $C_t = 100$ up to 100 iterations.

Table 6.1: Binary network V after minimizing the energy function for 85 iterations. Columnwise, first 12, then 10 and next 11 nodes represent feature primitives of 747, Airbus and DC10 aircraft symmetry sets of the model base respectively. Rows represent the nodes for features of figure 6.3.

000000000000	0000000000	01000000000
000000000000	0000000000	00010000000
000000000000	0000000000	00000010000
000000000000	0000000000	00000100000
000000000000	0000000000	00000000001
000000000000	0000000000	00000000010
000000000000	0000000000	00100000000
000000000000	0000000000	00000001000
000000000000	0000000000	00000000100

i.e. there are symmetry segments coinciding with the major axes of the shapes or x-axis (of the representative frame), only half of the symmetry set (the upper half of the x-axis) is used for graph matching.

We find the best match between the transformed symmetry set of figure 6.3 with one of the candidate models of figure 6.7. The energy function between the image and the integrated model graph is minimized using simulated annealing with constant values $A = B = C = D = G = 200$ and $F = 500$. This is to give more priority to the relational attribute matching since the symmetry set in the model base are similar structurally. All other weights are taken as unity for the current implementation. We have tested both the annealing schedule, given in equations (6.5) and (6.6). We observe that for both the cases, after approximately 85 iterations of minimization process, the network V becomes almost stable and reaches a "thermal equilibrium". Also the change in energy state (ΔE) becomes *insignificant*. The processing time including I/O is approximately 45 minutes in a SUN3 implementation. The matrix V after 85th iterations, shown in table 6.1, gives the perfect match between the symmetry sets of the image and the model.

This technique could identify the perfect match of a model (after 95 iterations of wave process) to an image of figure 6.2 which is generated only after 80 iterations of wave propagation algorithm. This, as claimed in the introduction, gives an advantage in reducing the processing time of the candidate image whose match is to be found.

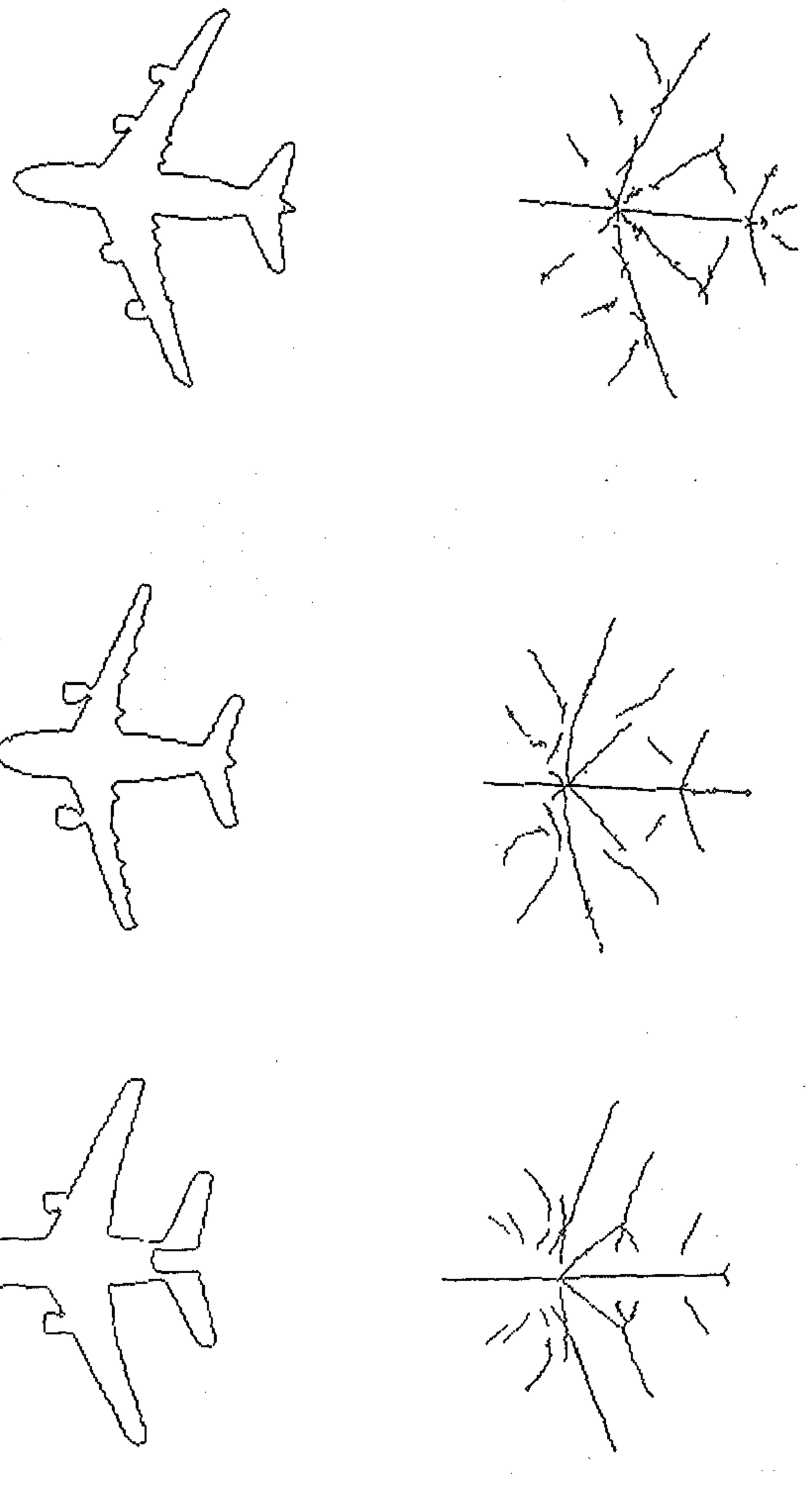

Figure 6.7: Model base consisting of symmetry set of 747, Airbus and DC10 aircraft.

6.5 Discussions

We have formulated a symmetry matching problem as minimization of an energy function. In the minimization process, both the local and global information are combined together and feature primitives of all the models are compared to that of the image at the same time. The conventional steps like hypotheses generation and verification could be excluded. This is more so because of the inter nodes cooperative voting and/or mutual inhibition (between the image versus different models) of the match of feature primitives. Because of stochastic behavior of simulated annealing process, chance of achieving global minima is higher. Though the process is cpu intensive, a faster alternative could be achieved by exploiting the inherent parallelism in the energy minimization process. We discuss this in the next chapter indicating the future directions and potentials of this research.

Chapter 7

Conclusion

7.1 Summary

We present a set of algorithms to detect and utilize local reflectional symmetry. In the first part, we have demonstrated that, despite the effect of "skewing" and "fanning" in the image plane due to affine or projective transformation of an object, bilateral symmetries could be detected using geometric invariance properties. These symmetries are then used to back-project images modulo similarity transform. This is achieved without using any model base or intrinsic camera parameters. In the next phase, the parallel implementation of the wave diffusion algorithm or the *normal transform* method provides almost all the *perceptually relevant* symmetry set for a planar shape under similarity transform. We finally utilize this as a cue for model based symmetry set matching. The structural and relational matching constraints give an energy function, which is optimised using simulated annealing.

To critically evaluate the computer vision algorithms developed in this thesis, we analyse their performances in the light of following desirable properties of shape description methodologies.

Speed A major failing of most current shape description algorithms is the time taken to compute even trivial shape descriptions. These lead to large amounts of computer time being necessary to generate a description sufficiently rich to enable inspection or recognition. The symmetry detection algorithms and

their uses, presented in this thesis, are fast and practically viable. In fact, a major aim of chapter 5 is to speed up the otherwise attractive process of symmetry detection using wave diffusion. The symmetry matching using simulated annealing is cpu intensive and thus far below the expectations so far as the speed is concerned. However, because of the inherent parallelism, the minimization of energy function using simulated annealing could be implemented in parallel hardware.

Robustness We have demonstrated performances of the algorithms with a number of images of real objects. Though, we admit that too much of "unpleasant" features or noises have been avoided, the local symmetry constraints we have utilized are quite applicable to real situations. Since, for symmetry detection, application and matching, we have utilized only the local information, the algorithms will expect to perform much better so far as noise robustness and occlusion are concerned.

Generality Generality is an important issue in shape description. We have shown our algorithms perform reliably for a variety of object classes. The real scene with multiple objects, some of which are even partially symmetric or non-symmetric, are taken as test objects. Since, the constraints for back-projection and the constraints for symmetry set matching assume no special pre-conditions, a reasonable amount of generality has been achieved. In case of symmetry matching, we do not claim this methodology to be rich and exhaustive enough to cover a wide variety of real scenes, the issues and limitations are being discussed in the next section. However, the process of building energy function is general enough to take different feature sets and optimization techniques.

Extendibility Though, algorithms for 2D shapes are certainly of value, the ultimate goal is to extend these ideas for 3D shapes. There are two definite possibilities of our contributions: The first approach is based on other object relations like exploiting rotational symmetries while the second one is extending these ideas to 3D shapes. We discuss all these issues in detail in the next section.

7.2 Future direction

1. Chapters 3 and 4

Applications of anti-symmetry set [8] in case of affine images and the semi-differential invariants [4,33] in case of projective images to detect local symmetry will be an important extension in combining invariance properties with the symmetry. This is particularly of importance, for images of convex shapes, which we have not dealt in detail.

Similar back-projection constraints can be derived for other object relations. For example a planar object with two-fold rotational symmetry induces the following constraints on the projectivity related corresponding image curves:

(a) $\mathrm{T}^2 = \mathrm{I}$

(b) $|\mathrm{T} + \mathrm{I}| = 0$

(c) $|\mathrm{T} - \mathrm{I}| \neq 0$

and similarly for n-fold rotational symmetry (with $\mathrm{T}^n = \mathrm{I}$), T being the matrix responsible for inter-symmetric-curve transformation.

Exploiting object specific heuristics, such as, fitting conic(s) (which back-projects to a circle) to a set of data points of a skewed scene, and its relationship to the *circular points*, $(1, i, 0)^t$ and $(1, -i, 0)^t$ may be used to unskew a "skewed" scene. This idea can also be extended for any object or its part with rotational symmetry.

Another extension is to affine/projective images of 3D objects with bilateral symmetry. Exploiting this constraint facilitates the recovery of 3D structure and pose from single images. It can be shown that, for affine views, structure can be recovered, modulo a Euclidean transformation, to a four parameter family of symmetric objects that could have given rise to the image [25]. If the object has two orthogonal bilateral symmetries, the shape can be reconstructed modulo similarity.

2. Chapter 5

The parallel implementation of wave diffusion process opens up a few key issues which need to be explored. The idea encapsulated in the parallelism of wave process is to find a parallel ridge finder of a blurred shape using Gaussian convolution by local iteration of a binomial mask. A simple hill-climbing technique may be implemented for this. Single instruction multiple data (SIMD) hardware could be a possibility for implementing such algorithms; however, communication overhead in this case may be a critical issue. A viable alternative may be an architecture like Connection Machine where every point in

the image plane has an associated processor. The idea of detecting symmetry in this case will be very similar to the *normal transform* method proposed in chapter 5. Reformulating wave propagation as message propagation, each processor will "watch" for two or more messages to "pass almost overhead" from different directions "almost simultaneously". Simulation of similar type of implementation is reported in [12].

3. Chapter 6

The potential of the shape matching process could be widened by investigating some of the limitations inherent in the wave diffusion process. For instance, the possible problem introduced when two sections of curve of very different lengths interact to produce symmetries and also, the behavior of the process when a lot of wave energy converges on the same spot - a potential rotationally symmetric point, which we have avoided in our analysis. Analysing only the linear and circular features have restricted the analysis. This may be extended to curve segments approximated with cubic b-spline. As an alternative to the transformation of symmetry set to representative frame, specially for noisy and occluded scene, the energy function may be upgraded to take care rotation, scaling and translation of planar shape. A thorough investigation needs to be done to evaluate the constants, weights and priorities of different costs of the energy function, and also a set of other annealing schedules, all of which influence the minimization process and the processing time. The minimization of energy function for matching using Hopfield net could as well be a possibility. Finally, a fruitful extension will be to implement this matching process on a parallel or neural hardware for real-time applications.

Appendix A

A.1 Proof of Theorem 1

Suppose two image curves γ and γ', as in figure 3.1, are images of two sides of a planar object with bilateral symmetry, and image projection can be represented by an affine transformation. Then the transformation between γ and γ' has the following properties:

1. γ and γ' are related by an affine transformation. That is, if $\mathbf{x}$ is a point on γ then there is a point $\mathbf{x}'$ on γ' such that:

$$\mathbf{x}' = \mathtt{A}\mathbf{x} + \mathbf{b}$$

where $\mathtt{A}$ is a non-singular 2×2 matrix, and $\mathbf{b}$ is a two-vector.

Proof The first statement is a straightforward consequence of the group closure property for affine transformations. Since the two sides Γ and Γ' of the object (see figure 3.1) are related by a reflection (which is affine), and since imaging is assumed to be affine, the two imaged sides γ and γ' are also related by an affine transformation.

In more detail, and to fix notation, object curves Γ and Γ' are related by a reflection, i.e. if $\mathbf{X}$ is a point on Γ then there is a point $\mathbf{X}'$ on Γ' such that:

$$\mathbf{X}' = \mathtt{S}\mathbf{X} + \mathbf{B}_S \tag{A.1}$$

where $\mathtt{S}$ is a reflection matrix ($|\mathtt{S}| = -1$). Under the image transformation equation (3.4)

$$\begin{aligned} \mathbf{X} &= \mathtt{U}\mathbf{x} + \mathbf{B} \\ \mathbf{X}' &= \mathtt{U}\mathbf{x}' + \mathbf{B} \end{aligned}$$

Combining these with equation (A.1) gives:

$$\mathbf{x}' = \mathtt{A}\mathbf{x} + \mathbf{b}$$

where

$$\begin{aligned} \mathtt{A} &= \mathtt{U}^{-1}\mathtt{S}\mathtt{U} \\ \mathbf{b} &= \mathtt{U}^{-1}(\mathtt{S}\mathbf{B} + \mathbf{B}_S - \mathbf{B}) \end{aligned}$$

□

2. The affine transformation $\{\mathtt{A}, \mathbf{b}\}$ obeys the following constraints:

   (a) $\mathtt{A}^2 = \mathtt{I}$
   (b) $|\mathtt{A} + \mathtt{I}| = 0$
   (c) $|\mathtt{A} - \mathtt{I}| = 0$

Proof The first part is straightforward:

$$\begin{aligned} \mathtt{A} &= \mathtt{U}^{-1}\mathtt{S}\mathtt{U} \\ \mathtt{A}^2 &= \mathtt{U}^{-1}\mathtt{S}^2\mathtt{U} \\ &= \mathtt{U}^{-1}\mathtt{U} \\ &= \mathtt{I} \end{aligned}$$

The image correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ relates points transformed according to the first part of the theorem. So $\mathbf{x}' = \mathtt{A}\mathbf{x} + \mathbf{b}$ and also $\mathbf{x} = \mathtt{A}\mathbf{x}' + \mathbf{b}$. Consequently, applying this transformation to a point $\mathbf{x}$ on γ maps it to a point $\mathbf{x}'$ on γ', and applying the transformation again maps it back to $\mathbf{x}$:

$$\begin{aligned} \mathbf{x} &= \mathtt{A}(\mathtt{A}\mathbf{x} + \mathbf{b}) + \mathbf{b} \\ &= \mathtt{A}^2\mathbf{x} + (\mathtt{A} + \mathtt{I})\mathbf{b} \end{aligned}$$

Since this is true for all $\mathbf{x}$ we have:

(a) Again $\mathtt{A}^2 = \mathtt{I}$
From this it follows that $\mathtt{A}$ has eigenvalues ± 1.

(b) $(\mathtt{A} + \mathtt{I})\mathbf{b} = 0$
Hence $|\mathtt{A} + \mathtt{I}| = 0$, since $\mathbf{b}$ is non-trivial, and $\mathbf{b}$ is an eigenvector of $\mathtt{A}$ with eigenvalue -1.

(c) If $\mathbf{x}$ is on the (imaged) symmetry axis then $\mathbf{x}' = \mathtt{A}\mathbf{x} + \mathbf{b} = \mathbf{x}$ and hence

$$(\mathtt{I} - \mathtt{A})\mathbf{x} = \mathbf{b} \tag{A.2}$$

This is true for all points on the axis. Consequently, $(\mathtt{A} - \mathtt{I})$ must be of rank 1 (i.e. $|\mathtt{A} - \mathtt{I}| = 0$) with the kernel of $(\mathtt{A} - \mathtt{I})$ defining the axis direction $\mathbf{a}$ i.e. $(\mathtt{A} - \mathtt{I})\mathbf{a} = 0$ which means that $\mathbf{a}$ is an eigenvector of $\mathtt{A}$ with eigenvalue 1. □

3. The matrix $\mathtt{A}$ has eigenvectors $\mathbf{a}$ and $\mathbf{b}$ with eigenvalues $+1$ and -1 respectively. Vector $\mathbf{a}$ is parallel to the symmetry axis, vector $\mathbf{b}$ is parallel to $\mathbf{x}' - \mathbf{x}$.

Proof All that remains to be shown is that $\mathbf{b}$ is parallel to $\mathbf{x}' - \mathbf{x}$. Note corresponding image points are joined by parallel lines since these lines are images of parallel lines on the object. Recall that the eigenvectors of $\mathtt{A}$ are $\mathbf{a}$ and $\mathbf{b}$ with eigenvalues $+1$, -1 respectively. $\mathbf{a}$ and $\mathbf{b}$ span the image, so that for some α, β:

$$\mathbf{x}' - \mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b}.$$

Applying $\mathtt{A}$ to both sides, and reversing the order,

$$\begin{aligned} \alpha\mathbf{a} - \beta\mathbf{b} &= \mathtt{A}\mathbf{x}' - \mathtt{A}\mathbf{x} \\ &= \mathbf{x} - \mathbf{x}' \\ &= -\alpha\mathbf{a} - \beta\mathbf{b} \end{aligned}$$

from which it follows that $\alpha = 0$, so that $\mathbf{x} - \mathbf{x}'$ is parallel to $\mathbf{b}$ as required. □

4. The transformation has three degrees of freedom. It can be parameterised by a, b_x, b_y where:

$$\mathtt{A} = \begin{bmatrix} a & -b_x(1+a)/b_y \\ -b_y(1-a)/b_x & -a \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \tag{A.3}$$

Proof Solving the equations $|\mathtt{A} + \mathtt{I}| = 0$ and $|\mathtt{A} - \mathtt{I}| = 0$ for $\mathtt{A}$ gives

$$\mathtt{A} = \begin{bmatrix} a & b \\ \frac{1-a^2}{b} & -a \end{bmatrix}$$

where a and b are two parameters. This also satisfies $\mathtt{A}^2 = \mathtt{I}$. The requirement that $\mathtt{A}\mathbf{b} = -\mathbf{b}$ determines b in terms of a, b_x and b_y.

The three parameters represent the symmetry line (two degrees of freedom) and the correspondence direction (one degree of freedom). □

5. With this notation, the image (skewed) symmetry line is: $(1-a)b_y x + (1+a)b_x y - b_x b_y = 0$.

Proof This is the particular solution of equation (A.2). □

A.2 Proof of Theorem 3

In the most general case there is a projective (rather than affine) transformation between object and image planes. As in the affine case symmetry in the object plane constrains the transformation between image curves.

Suppose two curves γ and γ', as in figure 4.3, are the images of two corresponding sides of a planar object with bilateral symmetry. Suppose further that image projection can be represented by a projective transformation. Then the transformation between γ and γ' has the following properties:

1. γ and γ' are related by a projective transformation. That is, if $\mathbf{x}$ is a point on γ then there is a point $\mathbf{x}'$ on γ' such that:

$$\mathbf{x}' = \mathtt{T}\mathbf{x} \tag{A.4}$$

where $\mathtt{T}$ is a non-singular 3×3 matrix, and $\mathbf{x}$ and $\mathbf{x}'$ are homogeneous three-vectors.

Proof This follows from group closure under projective transformations. If the object reflection is the projective transformation $\mathbf{X}' = \mathtt{S}\mathbf{X}$, and the object to image projection is

$$\begin{aligned} \mathbf{x} &= \mathtt{U}\mathbf{X} \\ \mathbf{x}' &= \mathtt{U}\mathbf{X}', \end{aligned}$$

then the transformation between γ and γ' is given by the conjugate projectivity:

$$\mathtt{T} = \mathtt{U}\mathtt{S}\mathtt{U}^{-1}. \square$$

2. The projective transformation T satisfies the following constraints:

   (a) $\mathtt{T}^2 = k\mathtt{I}$, where k is a scalar.

   (b) The fixed points of T are: a line of fixed points; and, a fixed point (not on the line) through which there is a pencil of fixed lines.

A projection with these properties is a collineation of period two, also known as a 2 cyclic homography, and a planar harmonic homology [69].

Proof

(a) We have the image correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ so that $\mathbf{x}' = \mathtt{T}\mathbf{x}$ and also $\mathbf{x} = \mathtt{T}\mathbf{x}'$. Consequently, applying this transformation to a point $\mathbf{x}$ on γ maps it to a point $\mathbf{x}'$ on γ', and applying the transformation again maps it back to $\mathbf{x}$:

$$\begin{aligned} \mathbf{x} &= \mathtt{T}(\mathtt{T}\mathbf{x}) \\ &= \mathtt{T}^2\mathbf{x} \end{aligned}$$

Since this is true for all $\mathbf{x}$, $\mathtt{T}^2 = k\mathtt{I}$.

(b) This can be proved analytically but is seen most simply by considering the projection geometry in figure 4.3. On the object, points on the symmetry axis are mapped by the reflection to themselves - so this is a line of fixed points. Correspondingly, the imaged symmetry line is a line of fixed points. The only other fixed point (not on this line) is the point at infinity (where parallel lines joining corresponding points intersect). In the image this is transformed to the vanishing point of the lines joining corresponding points. $\square$

3. The matrix T has eigenvectors $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Two of the eigenvalues, corresponding to $\mathbf{e}_2$ and $\mathbf{e}_3$ say, are equal. The third, corresponding to $\mathbf{e}_1$ is distinct and non-zero. The symmetry axis is given by the line $\mathbf{e}_2 \times \mathbf{e}_3$. Corresponding points, $\mathbf{x}'$ and $\mathbf{x}$, are collinear with $\mathbf{e}_1$. The line $\mathbf{x}'$, $\mathbf{x}$ intersects the symmetry axis in a point $\mathbf{x}_I$ say, and the four collinear points $\mathbf{x}$, $\mathbf{x}_I$ $\mathbf{x}'$ and $\mathbf{e}_1$ have a harmonic cross-ratio.

Proof For a line l of fixed points two degenerate eigenvectors must lie on the line. To see this represent a point on the line as $\mathbf{x} = \kappa\mathbf{e}_2 + \nu\mathbf{e}_3$. Then

$$\begin{aligned} \mathbf{x}' &= \mathtt{T}(\kappa\mathbf{e}_2 + \nu\mathbf{e}_3) \\ &= \lambda(\kappa\mathbf{e}_2 + \nu\mathbf{e}_3) \end{aligned}$$

(where $\mathbf{e}_2$ and $\mathbf{e}_3$ are eigenvectors of T with eigenvalue λ) which is the same point. Thus the symmetry line is given by $\mathbf{e}_2 \times \mathbf{e}_3$. The vanishing point corresponds to $\mathbf{e}_1$, the other fixed point. A similar argument to the above shows that any line passing through this point is a fixed line under T. On the object corresponding points are collinear with this point at infinity (since the lines joining corresponding points are all parallel), so in the image $\mathbf{x}'$ and $\mathbf{x}$, are collinear with $\mathbf{e}_1$.

On the object the four points $\mathbf{X}'$, $\mathbf{X}$, the intersection of their common line with the symmetry axis and the point at infinity have a harmonic cross-ratio. This is preserved by projectivities. Hence the four image points $\mathbf{x}'$, $\mathbf{x}$, $\mathbf{e}_1$ and the intersection of their common line with the symmetry axis, have a harmonic cross-ratio. □

4. The transformation has four degrees of freedom. It can be determined from two correspondences.

Proof These correspond to the symmetry axis (two degrees of freedom) and the vanishing point (two degrees of freedom). Two point correspondences, $\mathbf{x}_1 \leftrightarrow \mathbf{x}_1'$ and $\mathbf{x}_2 \leftrightarrow \mathbf{x}_2'$ gives four constraints:

$$\begin{aligned} \mathbf{x}_1' &= \mathtt{T}\mathbf{x}_1 \\ \mathbf{x}_2' &= \mathtt{T}\mathbf{x}_2 \\ \mathbf{x}_1 &= \mathtt{T}\mathbf{x}_1' \\ \mathbf{x}_2 &= \mathtt{T}\mathbf{x}_2' \end{aligned}$$

which is sufficient to determine T. □

Bibliography

[1] Leendert Ammeraal. *Computer Graphics.* John Wiley, UK, 1986.

[2] Kalle Astrom. *Affine Invariants of Planar Sets.* Technical Report Report No. 7005, Lund Institute of Technology, Department of Mathematics, 1992.

[3] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentic-Hall Inc., USA, 1982.

[4] E. B. Barett, P. M. Payton, and M. H. Brill. Contributions to the theory of projective invariants for curves in two and three dimensions. In *First DARPA-ESPIRIT Workshop on Invariance*, pages 325–328, 1991.

[5] J. Basak, S. Chaudhury, S. K. Pal, and D. D. Majumder. Matching of structural shape descriptions with hopfield net. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(2):377–404, 1993.

[6] I. Beiderman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[7] A. Blake and C. Marinos. Shape from texture: estimation, isotropy and moments. *Artificial Intelligence*, 45:323–380, 1990.

[8] A. Blake and M. Taylor. Grasping visual symmetry. In *Proceedings of 4th International Conference on Computer Vision*, pages 724–733, 1993.

[9] H. Blum. Biological shape and visual science (part i). *Journal of Theoretical Biology*, 38:205–287, 1973.

[10] H. Blum and R. N. Nagel. Shape description using symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

[11] J. M. Brady and H. Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61, 1984.

[12] J. M. Brady and G. L. Scott. Parallel algorithms for shape representation. In *Parallel Architectures and Computer Vision*, pages 97–118, Oxford University Press, Oxford, UK, 1988.

[13] J. M. Brady and A. Yuille. An extremum principle for shape from contour. *PAMI*, 6:288–301, 1984.

[14] R. A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348, 1981.

[15] J. F. Canny. Finding edges and lines in images. *PAMI*, 8(6):679–698, 1986.

[16] J. F. Canny and B. R. Donald. Simplified voronoi diagrams. In *ACM Symposium on Computational Geometry*, 1987.

[17] J. H. Connell and J. M. Brady. Generating and generalising models of visual objects. *Artificial Intelligence*, 31:159–184, 1987.

[18] M. C. Corballis. Recognition of disoriented shapes. *Psychological Review*, 95:115–123, 1988.

[19] J. L. Crowley and A. C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE PAMI*, 6(2):156–170, 1984.

[20] Lawrence Davis. *Genetic Algorithms and Simulated Annealing*. Morgan Kaufmann Publishers, London, UK, 1987.

[21] P. Eades. Symmetry finding algorithms. In *Computational Morphology*, pages 41–51, Elsevier Science Publishers B.V., North-Holland, 1988.

[22] P. Eades and A. Lee. *Perception of Symmetry*. Technical Report Report No. 52, University of Queensland, Department of Computer Science, 1985.

[23] Olivier Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint*. MIT Press, Boston, USA, 1993.

[24] I. D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture*. John Wiley, UK, 1986.

[25] R. Fawcett, A. Zisserman, and J. M. Brady. Extracting structure from single affine views of 3d point sets with one or two bilateral symmetries. In *Proceedings of British Machine Vision Conference*, 1993.

[26] M. Field and M. Golubitsky. *Symmetry in Chaos: A Search for Pattern in Mathematics, Art and Nature.* Oxford University Press, Oxford, UK, 1992.

[27] M. Fleck. *Local Rotational Symmetries.* Technical Report Report No. AI-TR-852, MIT, Boston, USA, 1985.

[28] D. Forsyth, J. L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant description for 3-d object recognition and pose. *IEEE PAMI*, 13(10):971–991, 1991.

[29] H. Freeman. Computer processing of line drawing images. *Computer Surveys*, 6(1):57–98, 1974.

[30] S. A. Friedberg. Finding axes of skewed symmetry. *Computer Vision, Graphics and Image Processing*, 34(2):138–155, 1986.

[31] Jonas Garding. *Shape from Surface Markings.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1991.

[32] P. J. Giblin and S. A. Brassett. Local symmetry of plane curve. *American Mathematical Monthly*, 92(10):689–707, 1985.

[33] L. J. Van Gool, T. Moons, E. Pauwels, and A. Oosterlinck. Semi differential invariants. In *Geometric Invariance in Computer Vision*, pages 157–192, MIT Press, Boston, USA, 1992.

[34] W. E. L. Grimson. *Object Recognition by Computer - The Role of Geometric Constraints.* MIT Press, Boston, USA, 1990.

[35] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision.* Addison-Wesley Publishing Company, NY, USA, 1992.

[36] R. Horaud and J. M. Brady. On the geometric interpretation of image contours. *Artificial Intelligence*, 37:333–353, 1988.

[37] T. Kanade. Recovery of three dimensional shape of an object from a single view. *Artificial Intelligence*, 17:409–460, 1981.

[38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization using simulated annealing. *Science*, 220:671–680, 1983.

[39] Y. Lamdan, J. Schwartz, and H. Wolfson. Object recognition by affine invariant matching. In *Proceedings of Computer Vision and Pattern Recognbition*, 1988.

[40] M. Leyton. A process-grammar for shape. *Artificial Intelligence*, 34:213–247, 1988.

[41] Ja-Chen Lin. Universal principal axes: an easy-to -construct tool useful in defining shape orientations for almost every kind of shape. *Pattern Recognition*, 26(4):485–493, 1993.

[42] D. G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, Boston, USA, 1985.

[43] D. Marr. *Vision.* Freeman, New York, USA, 1982.

[44] S. Marshall. Review of shape coding techniques. *Image and Vision Computing*, 7(4):281–294, 1989.

[45] R. Mohan and R. Nevatia. Perceptual organization for segmentation and description. In *Proceedings of DARPA Image Understanding Workshop*, pages 415–424, 1989.

[46] U. Montanari. A method for obtaining skeletons using a quasi-euclidean distance. *Journal of the ACM*, 15:600–624, 1968.

[47] J. C. Mott-Smith. Medial axis transformations. In *Picture Processing Psychopictorics*, Academic Press, NY, USA, 1970.

[48] D. P. Mukherjee and J. M. Brady. Symmetry analysis through wave propagation. 1993. Communicated to International Journal of Pattern Recognition and Artificial Intelligence.

[49] D. P. Mukherjee and D. Dutta Majumder. Implementing wave propagation on transputers for symmetry detection. In *Proceedings of 3rd International Conference on Advances in Pattern Recognition and Digital Techniques, Calcutta*, pages 149–158, 1993.

[50] D. P. Mukherjee and D. Dutta Majumder. Shape symmetry matching using simulated annealing. 1994. Communicated to IEEE Transactions on Systems, Man and Cybernetics.

[51] D. P. Mukherjee and A. Zisserman. Shape from symmetry - detecting and exploiting symmetry in projective images. 1994. An intermediate draft.

[52] D. P. Mukherjee, A. Zisserman, and J. M. Brady. Shape from symmetry - detecting and exploiting symmetry in affine images. 1993. Accepted in Proc. Royal Society, Series A.

[53] J. L. Mundy and A. Zisserman(Ed.). *Geometric Invariance in Computer Vision*. MIT Press, Boston, USA, 1992.

[54] N. M. Nasrabadi, Li Wei, and Chang Y. Choo. Object recognition by hopfield neural network. In *ICCV'90*, pages 325–328, 1990.

[55] S. M. Pizer, W. R. Oliver, and S. H. Bloomberg. Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE PAMI*, 9(4):505–511, 1987.

[56] J. Ponce. Ribbons, symmetries, and skew symmetries. In *Proceedings of DARPA Image Understanding Workshop*, pages 1074–1079, 1988.

[57] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, UK, 1993.

[58] I. Reid. *Recognising Parameterized Models from Range Data*. PhD thesis, University of Oxford, Oxford, UK, 1991.

[59] H. Rom and G. Medioni. Hierarchical decomposition and axial representation of shape. *PAMI*, 15(10):973–981, 1993.

[60] A. Rosenfeld. Axial representations of shape. *Computer Vision, Graphics and Image Processing*, 33:156–173, 1986.

[61] A. Rosenfeld and J. L. Pflatz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–62, 1968.

[62] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Canonical frames for planar object recognition. In *Proceedings of ECCV2*, pages 757–772, 1992.

[63] C. A. Rothwell, A. Zisserman, C. Marinos, D. A. Forsyth, and J. L. Mundy. Relative motion and pose from arbitrary plane curves. *Image and Vision Computing*, 10(4):250–262, 1992.

[64] C. A. Rothwell, A. Zisserman, J. L. Mundy, and D. A. Forsyth. Efficient model library access by projectively invariant indexing. In *Proceedings of Computer Vision and Pattern Recognition*, pages 109–114, 1992.

[65] G. L. Scott, S. Turner, and A. Zisserman. Using a mixed wave/diffusion process to elicit the symmetry set. *Image and Vision Computing*, 7(1):63–70, 1989.

[66] Gong Shaogang. *Parallel Computation of Visual Motion*. PhD thesis, University of Oxford, Oxford, UK, 1989.

[67] L. G. Shapiro and R. M. Haralick. Organisation of relational models for scene analysis. *IEEE PAMI*, 4(6):595–602, 1982.

[68] K. Y. Shiu-Yin. *Shape from Contour Using Symmetries.* Technical Report Report No. CSRP 141, University of Sussex, Sussex, UK, May, 1989.

[69] C. E. Springer. *Geometry and Analysis of Projective Spaces.* Freeman, New York, USA, 1964.

[70] G. Stephenson. *An Introduction to Partial Differential Equations for Science Students.* Longmans, London, UK, 1968.

[71] K. A. Stevens. Slant-tilt: the visual encoding of surface orientation. *Biological Cybernetics*, 46:183–195, 1983.

[72] N. J. Strachan. Recognition of fish species by colour and shape. *Image and Vision Computing*, 11(1):2–10, 1993.

[73] A. Thornham, C. J. Taylor, and D. H. Cooper. Object cues for model-based image interpretation. In *Fourth Alvey Vision Conference*, 1987.

[74] F. Ulupinar and R. Nevatia. Recovery of 3-d objects with multiple curved surfaces from 2-d contours. 1993. An intermediate draft.

[75] J. Wagemans. Skewed symmetry: a nonaccidental property used to perceive visual forms. *Journal of Expert Psychology: Human Perception and Performance*, 19(2):1–17, 1993.

[76] Isaac Weiss. *Geometric Invariants and Object Recognition.* Technical Report Report No. CAR-TR-632, University of Maryland, Centre for Automation Research, 1992.

[77] H. Weyl. *Symmetry.* Princeton University Press, Princeton, USA, 1952.

[78] A. P. Witkin and J. M. Tenenbam. On the role of structure in vision. In *Human and Machine Vision*, pages 481–543, Academic Press, NY, USA, 1983.